

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: «Дослідження інформаційних потоків та розробка підсистеми автоматизації документообігу в інформаційній системі ВП Південноукраїнська АЕС»

Виконав: студент групи 6142 м

_____ ***Білик О.А.***
(підпис)

Керівник роботи:

старший викладач
(посада, науковий ступінь вчене звання)
_____ ***Морозова Г.С.***
(підпис)

м. Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"
Гарант освітньої програми

к.т.н, доц. Ажищев В. Ф.
" ____ " _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Білику Олександрю Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження інформаційних потоків та розробка підсистеми автоматизації документообігу в інформаційній системі ВП Південноукраїнська АЕС

Керівник роботи Морозова Ганна Сергіївна _____

Затверджені наказом ректора № 1098 -уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Морозова Г.С.	05.09.2023	22.12.2023
2	Морозова Г.С.	23.12.2023	15.06.2023
3	Морозова Г.С.	15.06.2023	25.09.2023
4	Морозова Г.С.	25.09.2023	14.11.2023
5	Морозова Г.С.	25.09.2023	14.11.2023

7. Дата видачі завдання 05 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2023	
2	Розробка концепції	15.03.2023	
3	Розділ з розробки проектних рішень по системі (та її частинах)	15.06.2023	
4	Реалізація проекту та розробка робочої документації	25.09.2023	
5	Вирішення питань охорони праці та навколишнього середовища	14.11.2023	
6	Оформлення пояснювальної записки	14.11.2023	
7	Подання магістерської роботи на перевірку на плагіат	16.11.2023	
8	Подання магістерської роботи рецензенту	20.11.2023	
9	Подання магістерської роботи на державний захист	15.12.2023	

Студент

_____ (підпис)

Керівник роботи

_____ (підпис)

Білик О.А.

(ПІБ)

Морозова Г.С.

(ПІБ)

АНОТАЦІЯ

Дипломна робота присвячена дослідженню інформаційних потоків та розробці проекту інформаційної управляючої системи документообігу ВП Південноукраїнська АЕС з метою підвищення продуктивності й поліпшення якості її документообігу.

В роботі сформовані вимоги до системи, розроблені її концепції і проведено вибір найкращої системи. Дипломна робота містить проектну, технічну та експлуатаційну документацію, а також розділ з охорони праці.

Дипломна робота викладена на 102 друкованих сторінках й містить: 15 рисунків, 17 таблиць, 4 додатка та список літератури з 10 найменувань.

ABSTRACT

The thesis is devoted to the study of information flows and the development of the project of the information management system of document flow VP South Ukrainian NPP with the aim of increasing productivity and improving the quality of its document flow.

In the work, requirements for the system were formed, its concepts were developed, and the best system was selected. The thesis contains design, technical and operational documentation, as well as a section on labor protection.

The thesis is laid out on 102 printed pages and contains: 15 figures, , 17 tables, , 4 appendices and a bibliography of 10 items.

ЗМІСТ

ВСТУП

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ТА ФОРМУВАННЯ ВИМОГ ДО АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОКУМЕНТООБІГУ ВП ПІВДЕННОУКРАЇНСЬКА АЕС

- 1.1 Формування вимог до автоматизованої системи управління
 - 1.1.1 Характеристики об'єкта і результати його діяльності
 - 1.1.2 Опис діючої інформаційної системи підприємства
- 1.2 Формування вимог до інформаційної системи
 - 1.2.1 Опис недоліків існуючої системи
 - 1.2.2 Аналіз існуючих рішень
 - 1.2.3 Основні елементи системи
 - 1.2.4 Цілі і критерії створюваної автоматизованої інформаційної системи
 - 1.2.5 Функції та задачі створюваної системи документообігу
- 1.3 Розробка концепції інформаційної системи
 - 1.3.1 Загальні вимоги до інформаційної системи
 - 1.3.2 Опис результатів вивчення об'єктів
 - 1.3.3 Варіант концепції інформаційної системи на базі FTP сервера
 - 1.3.4 Варіант концепції «Система управління документами»
- 1.4 Функціонально-вартісний аналіз
- 1.5 Оцінка надійності
- 1.6 Постановка задачі
- 1.7 Умови приймання системи

2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМІ ДОКУМЕНТООБІГУ ВП ПІВДЕННОУКРАЇНСЬКА АЕС

- 2.1 Загально-системні проектні рішення
 - 2.1.1 Організаційна і функціональна структура підприємства
 - 2.1.2 Загальний опис інформаційної системи
- 2.2 Рішення по організаційному забезпеченню
 - 2.2.1 Організаційна структура

- 2.2.2 Опис організації структури підприємства
- 2.2.3 Організаційно-кадрові заходи
- 2.3 Робочий проект
- 2.3.1 Рішення з інформаційного забезпечення
- 2.3.2 Опис інформаційного забезпечення
- 2.3.3 Опис організації інформаційної бази даних
- 2.3.4 Логічна модель даних
- 2.3.5 Фізична модель даних
- 2.4 Рішення з технічного забезпечення
- 2.5 Рішення з програмного забезпечення
- 2.5.1 Розробка діаграми класів програмного забезпечення
- 2.5.2 Розгортання програмного забезпечення

3 РОЗРОБКА РОБОЧОЇ ДОКУМЕНТАЦІЇ СИСТЕМИ УПРАВЛІННЯ ДОКУМЕНТООБІГУ ВП ПІВДЕННОУКРАЇНСЬКА АЕС

4 ОХОРОНА ПРАЦІ

- 4.1 Вступ
- 4.2 Аналіз небезпечних і шкідливих виробничих факторів
- 4.3 Розрахунок системи освітлення для приміщення
- 4.4 Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером

5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

ДОДАТОК Б – КЕРІВНИЦТВО КОРИСТУВАЧА

ДОДАТОК В – ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ

ДОДАТОК Г – ТЕКСТ ПРОГРАМИ

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ТА ФОРМУВАННЯ ВИМОГ ДО АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОКУМЕНТООБІГУ ВП ПІВДЕННОУКРАЇНСЬКА АЕС

1.1 Формування вимог до автоматизованої системи управління

1.1.1 Характеристики об'єкта і результати його діяльності

Основною метою діяльності ВП ЮУ АЕС є безпечне та ефективне виробництво електричної і теплової енергії, забезпечення умов для безпечної експлуатації енергоблоків АЕС і гідроагрегатів Каскаду ГЕС-ГАЕС.

Однією з умов досягнення цієї мети є підтримка на належному рівні системи забезпечення якості, яка вирішує такі основні завдання:

- створення, впровадження, використання в діяльності ефективної системи управління, що забезпечує оптимальне виконання робіт та процесів з безпечної виробленні електроенергії;
- забезпечення гарантій безпеки і демонстрація здатності до забезпечення рівня безпеки, відповідного вимогам правил, норм і стандартів з безпеки, а також умовам виданих ліцензій і дозволів;
- досягнення впевненості в тому, що вимоги діючих норм, правил і стандартів з безпеки та якості при здійсненні діяльності виконуються;
- залучення всього персоналу в процес поліпшення якості з чітким встановленням повноважень і відповідальності в сфері якості;
- планування та постійне вдосконалення системи забезпечення якості, виходячи з покладених зобов'язань і наявних / очікуваних ресурсів.

У ВП ПУАЕС всі структурні підрозділи беруть участь у реалізації системи управління якістю, причому встановлюється наступний розподіл функцій:

- керівництво ВП ПУАЕС забезпечує планування, організацію, управління, ресурси;
- особи, які виконують роботу, забезпечують якість;
- особи, що здійснює оцінку, аналізують якість і ефективність процесів управління та виконання роботи.

Організаційна структура ВП ПУАЕС, функціональні обов'язки, рівні повноважень та взаємодія керівників і персоналу, включаючи взаємодії між підрозділами, чітко визначені і документально оформлені. Для розробки і впровадження системи якості керівництво ВП ПУАЕС:

- встановило цілі, які повинні бути досягнуті в результаті діяльності підприємства, принципи і способи їх досягнення;
- сформулювало перелік нормативної та виробничої документації, яким має керуватися ВП ПУАЕС (див. Рисунок 1.1, Таблиця 1.1);
- розподілило функції, відповідальність і повноваження між підрозділами та керівниками ВП ПУАЕС;
- визначило види діяльності (роботи), що впливають на безпеку і оцінило фактичний стан забезпечення якості цих робіт відповідно до вимог забезпечення якості.



Рис. 1.1 – Документація, якою керується ВП ПУАЕС

Таблиця 1.1 - Склад ієрархічних рівнів документації

Нормативно-правові документи органів влади	
•	Закони України;

• Постановлення та Дикрети КМУ; • Укази и Постановления Президента Украины.		
Нормативні документи зовнішніх організацій:		
• державні та міждержавні стандарти; • нормативні документи вищого органу державного управління (галузеві стандарти, норми, правила, типові інструкції тощо); • нормативні документи органів державного регулювання та нагляду; • міжгалузеві нормативні документи інших міністерств і відомств; • технічні умови на продукцію, зареєстровані в органах Держстандарту.		
Документи Компанії	Документи ВП ПУАЕС	Документи структурних підрозділів
0	1	2
Основоположні документи:		
• Статут підприємства • Положення про експлуатуючої організації (оператора) ядерних установок «Національна атомна енергогенеруюча компанія «Енергоатом »» • Загальне керівництву за якістю ДП НАЕК «Енергоатом»	• Положення про ВП ПУАЕС ДП НАЕК «Енергоатом» • Загальне керівництво за якістю • ВП ПУАЕС • Технологічний регламент безпечної експлуатації; - Положення про розподіл напрямків діяльності в керівництві	

Керівні документи:		
• Положення Компанії про організацію діяльності • СТП Компанії • Посадові інструкції віце-президентів Компанії - Програми якості Компанії за напрямками діяльності	• Положення ВП ПУАЕС про організацію діяльності • СТП ВП ПУАЕС • Посадові інструкції керівників вищої ланки управління ВП ПУАЕС; • Програми (керівництва) якості ВП ПУАЕС за напрямками діяльності - Інструкції ВП ПУАЕС з безпеки (ядерної, радіаційної, пожежної)	• Положення про підрозділ Посадові інструкції, кваліфікаційні характеристики персоналу структурних підрозділів
Робочі документи:		
• Типові інструкції, методики, процедури • Перспективні та поточні плани робіт Компанії • Плани аварійного реагування Компанії - Переліки документації Компанії	• Загальностанційні інструкції, методики, програми виконання робіт • Інструкції з ліквідації аварійних ситуацій і аварій у ВП ПУАЕС і реакторних установках • Плани, графіки виконання робіт у • ВП ПУАЕС • Оперативні плани пожежогасіння	• Інструкції, програми, методики підрозділів • Плани, графіки виконання робіт підрозділів • Карти пожежогасіння • Переліки документації підрозділів - Переліки

	• Переліки документації ВП ПУАЕС - Схеми, креслення систем і обладнання ВП ПУАЕС	документації на робочих місцях персоналу
Записи:		
• акти (випробувань, приймання обладнання і т.д.); • протоколи (вимірювань, перевірки знань і т.д.); • звіти; • свідоцтва (про монтаж, виготовленні і т.д.); • сертифікати (відповідності продукції встановленим вимогам); журнали (оперативні, контролю і т.д.).		

Обов'язковими елементами системи управління якістю ВП ПУАЕС є:

- Організаційна структура.
- Підготовка та кваліфікація персоналу.
- Управління невідповідностями та коригувальні дії.
- Управління документами та записами.
- Виконання робіт.
- Управління проектуванням.
- Управління поставками.
- Контроль та випробування.
- Самооцінка керівництва.
- Незалежна оцінка.

В даний час ВП ПУАЕС має наступну структуру, зображену на Рис. 1.2.

Незважаючи на те, що всі підрозділи компанії мають схожу організаційну структуру, кожний з підрозділів (відділів, цехів) спеціалізується на виконанні проектів в своїй області (економіка і фінанси, забезпечення виробництва, фізичного захисту, якості управління і т.д.).

Генеральний директор (головний офіс) - за допомогою сектора помічників управляє всією компанією, працює з замовниками, розподіляє проекти між підрозділами, укладає договори, контролює виконання проектів на всіх стадіях, включаючи документообіг.

Фінансовий департамент (бухгалтерія, фінансове управління, планово-економічне управління) - займається розподілом, обробкою та обліком фінансових ресурсів.

Департамент з капітального будівництва - займається управлінням будівництвом об'єктів на території, монтажними роботами як на території підприємства, так і за його межами, управлінням по новим проектам та інвестиціям.

Департамент по забезпеченню виробництва - займається виробничо-технічної комплектацією, автотранспортним господарством, господарським обслуговуванням, ремонтом і обслуговуванням об'єктів СКБ.

Департамент з кадрів і соціального розвитку - займається підбором кадрів, соціальним розвитком, охороною здоров'я, зв'язками з громадськістю та ЗМІ.

Департамент з фізичного захисту та режиму - забезпечує дотримання режиму секретності і спеціального зв'язку, фізичний захист, відомчу воєнізовану охорону.

Департамент з якості та управління - забезпечує контроль якості, ліцензування, своєчасного оновлення та впровадження інформаційних технологій, налагодження міжнародного співробітництва та технічної допомоги, технічний контроль.

Организационная структура управления обособленного подразделения
"Южно-Украинская АЭС"

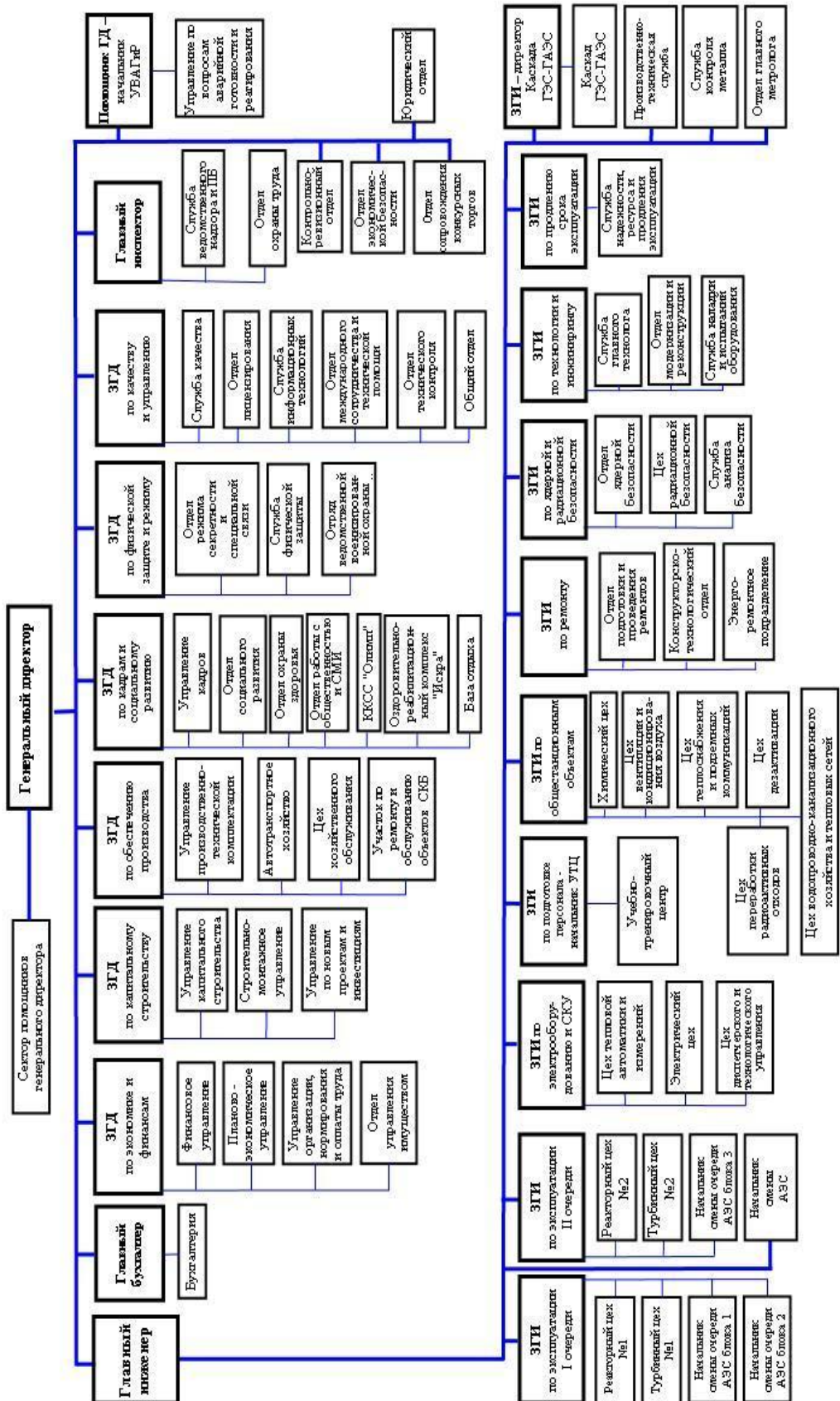


Рис. 1.2 – Організаційна структура ВП ПУАЕС

1.1.2 Опис діючої інформаційної системи підприємства

З Рисунка 1.3 бачимо, що в підрозділі і в кожному департаменті є комп'ютерне та офісне обладнання (принтери та факси), що дозволяє обмінюватися інформацією. Офіс (адміністрація) і всі департаменти, підрозділи між собою пов'язані. При розробці проектів з'являються набори документів, які бувають паперовими та електронними і зберігаються в різних архівах (паперовий і електронний архіви).

В паперовий архів здається вся документація усіма департаментами підрозділу. Кожен департамент має право доступу тільки до своєї документації. У зв'язку із значною кількістю проектів та документів за проектами необхідно витратити багато часу для дозволу на доступ та пошук потрібної інформації. На підприємстві існує автоматизована система управління проектами, але при цьому немає автоматизованої системи управління документами, яка могла б вирішити проблему доступу до документів, тому в електронному архіві вся доступна інформація зберігається на різних носіях (для кожного департаменту окремо).

У кожному департаменті є свій набір програмного забезпечення:

- клієнт електронної пошти - для обміну документами;
- інтернет - пошук необхідної документації, ГОСТів і стандартів;
- пошук нових технологій та пропозицій, дослідження ринку необхідних проектів та обмін інформацією між підрозділами;
- Скайп - проведення відео конференцій між підрозділами та Головним офісом, обмін інформацією;
- автоматизована система управління документами.

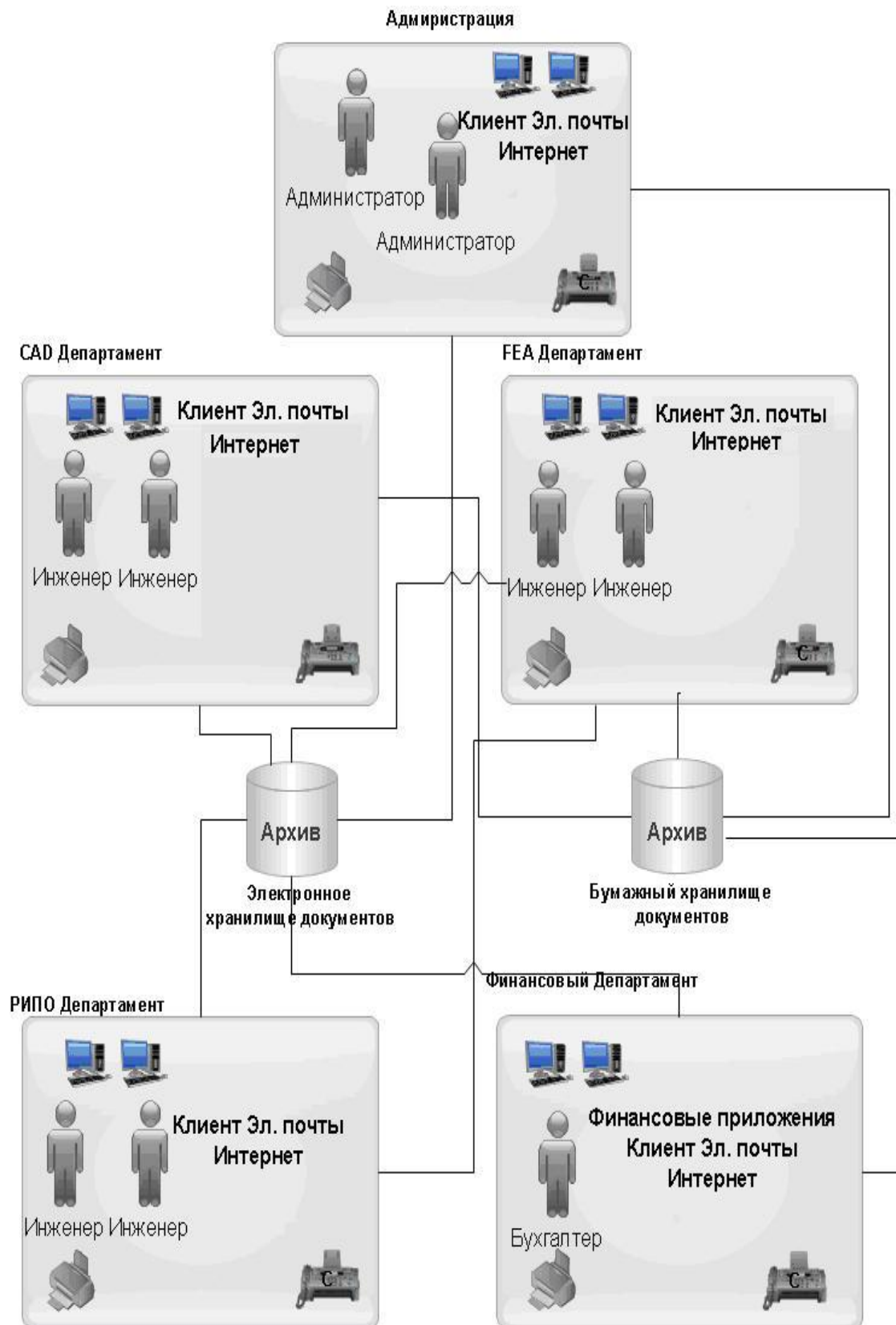


Рисунок 1.3 – Структурна модель діючої інформаційної системи ВП ПУАЕС

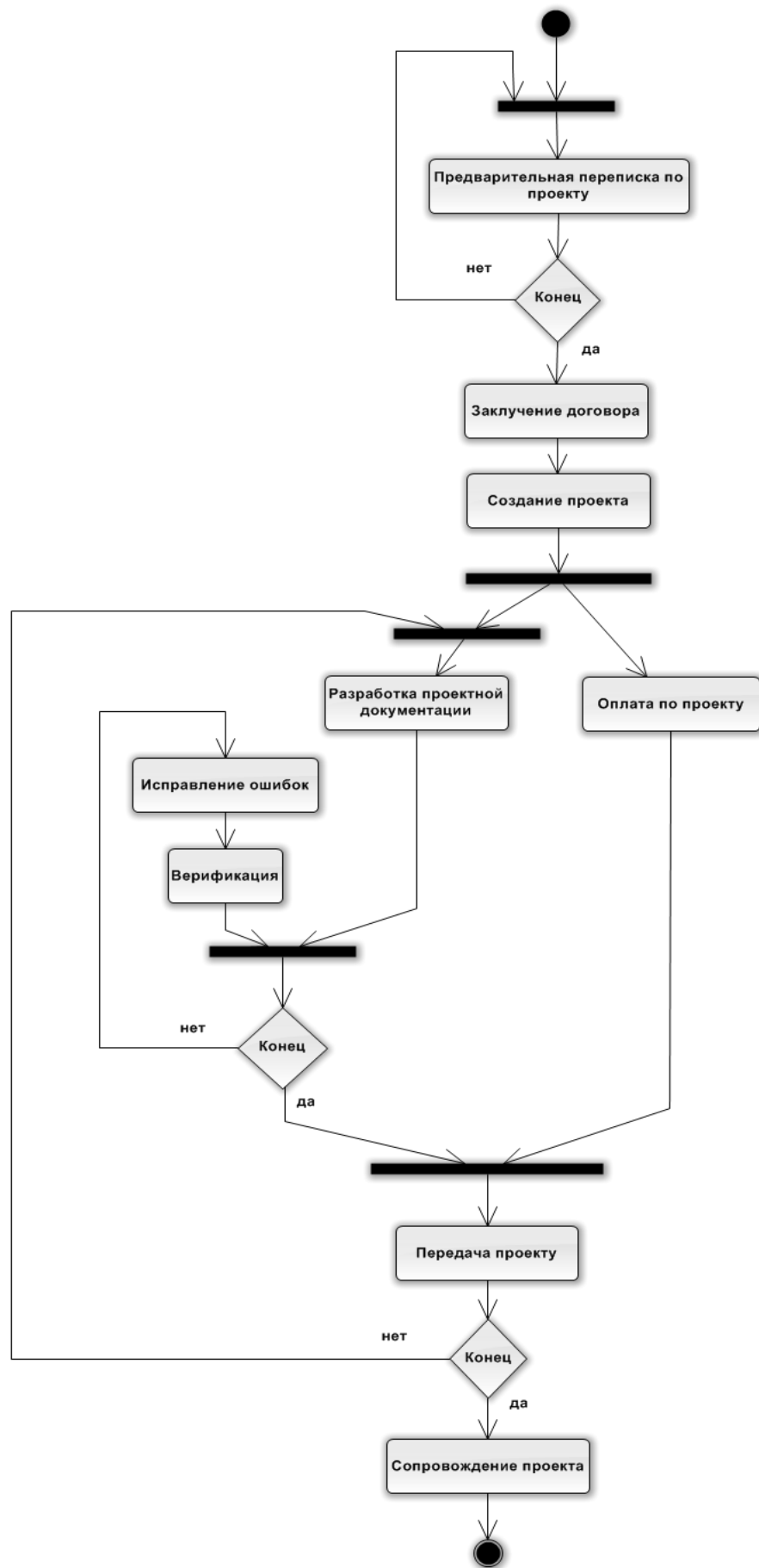


Рисунок 1.4 – Діаграма діяльності підприємства
У фінансовому департаменті з моменту укладення договору поетапно проводиться оплата відповідно до складеного договору.

Жоден з представлених об'єктів бізнес-процесу не автоматизовано, тобто всі дії з надійшли замовленням клієнта здійснюються або в усному спілкуванні, або на паперовому носії, отже, висока ймовірність того, що паперові документи або будуть загублені повністю, або частина з них буде втрачена в силу людського фактору.

1.2 Формування вимог до інформаційної системи

1.2.1 Опис недоліків існуючої системи

На підприємстві багато замовників і проектів, у зв'язку з цим багато документації, яку необхідно обробляти, зберігати і передавати. І в той же час є документи, які необхідно розробляти спільно кільком департаментам одночасно, а пошук необхідного документа займає багато часу.

Так само є недоліком те, що один і той же документ, отриманий на обробку з двох або більше департаментів, може бути оформлений у різних версіях або форматах.

Недоліки процесу документообігу, пов'язані з діяльністю підприємства є загальними для випадку, коли цей процес автоматизується фрагментарно без систематизації на рівні всього підприємства. Ці недоліки такі:

- низька швидкість обміну документацією;
- великі витрати на передачу інформації, документації;
- зберігання одних і тих самих документів в різних місцях у кількох видах, що викликає загрозу втрати актуальності та достовірності, а також повторення операцій з обробки;
- висока можливість спотворення і втрати документів при їх передачі.

На практиці це призводить до того, що при роботі з документацією виникає необхідність надмірного спілкування керівників усіх рівнів між собою і підлеглими, як безпосередньо, так і за допомогою паперових документів, з метою підтримки інформації про процеси виробництва в актуальному стані.

1.2.2 Аналіз існуючих рішень

На сьогоднішній день на ринку програмного забезпечення існує набір інформаційних систем , які могли б вирішити частину або повністю існуючі проблеми на підприємстві ВП «Південноукраїнська АЕС» , такі як:

DOCUMETR for SharePoint - це потужна система електронного документообігу на базі Microsoft SharePoint , яка включає всі необхідні функції по спільній роботі з документами і завданнями і дозволяє автоматизувати будь-які бізнес - процеси , пов'язані з рухом документів і виконанням завдань . Це система реально підвищує швидкість і продуктивність бізнес- процесів у компанії , допомагає налагодити узгоджену роботу підрозділів і включає в себе електронний архів документів , веде всю історію зміни завдань і документів , надає широкі можливості наскрізного пошуку інформації по всіх документах і завданням , а так само підтримує роботу з сканером і факсом , дозволяючи автоматизувати рутинні операції .

Newinttech - Діловодство:

Newinttech - Діловодство це безкоштовне прикладне рішення на базі IBM Lotus Notes / Domino " Newinttech Діловодство" . яке дозволяє автоматизувати роботу з вхідними , вихідними , внутрішніми та організаційно -розпорядчими документами компанії і забезпечує повну автоматизацію роботи канцелярії.

" Newinttech : Діловодство" містить наступні бази:

- внутрішні;
- вхідні;
- вихідні;
- контроль завдань;
- кореспонденти;
- номенклатура справ;
- ознайомлення;
- ОРД;
- проекти документів;
- узгодження;

- довідка;
- структура організації;
- структура системи;
- повідомлення;
- формування справ.

DocsVision:

Система електронного документообігу DocsVision призначена для створення гнучких, ефективних рішень:

- електронних архівів,
- автоматизації діловодства та контролю виконавської дисципліни;
- автоматизації документообігу – управління діловими та організаційними бізнес-процесами;
- інтеграції різних інформаційних систем підприємства на базі наскрізних бізнес-процесів.

DocsVision включає в себе наступні окремо поставлені компоненти:

- платформу DocsVision з вбудованими додатками, засобами настройки і розробки рішень;
- додатки – прикладні рішення, побудовані на платформі DocsVision і реалізують спеціалізовану функціональність для класу задач предметної області системи;
- додаткові модулі – технологічні компонентт, що реалізують розширення інтеграційних можливостей системи , а також розширюють можливості масштабування системи.

Таблиця 1.1 Аналіз функціональності існуючих рішень

	УД	СП	ОВПД	ПЗ
DOCUMETR	Так	Так	Hi	Windows Server
Newinttech	Так	Hi	Hi	Windows/Unix

DocsVision	Так	Ні	Ні	Windows/Unix
------------	-----	----	----	--------------

УД: управління доступом;

СП: можливістю документального супроводу історії проекту;

ОВПД: обліку версій проектної документації;

ПЗ: платформна залежність.

Дослідивши існуючі рішення видно , що вони не дозволяють комплексно вирішувати нашу задачу:

- немає можливості документального супроводу історії проекту;
- немає обліку версії проектної документації;
- немає можливості інтегрувати ці системи з існуючою інформаційною системою.

1.2.3 Основні елементи системи

Документообіг — рух документів в організації з моменту їх створення або отримання до завершення виконання або відправлення (ГОСТ Р 51141-98); комплекс робіт з документами: прийом, реєстрація, розсилка, контроль виконання, формування справ, зберігання та повторне використання документації , довідкова робота.

Електронний документообіг (ЕДО) — єдиний механізм по роботі з документами , представленими в електронному вигляді , з реалізацією концепції « безпаперового діловодства ».

Електронний документ (ЕД) — документ, створений за допомогою засобів комп'ютерної обробки інформації, який може бути підписаний електронним цифровим підписом (ЕЦП) і збережений на машинному носії у вигляді файла відповідного формату.

Електронний цифровий підпис (ЕЦП) — аналог власноручного підпису, що є засобом захисту інформації, що забезпечує можливість контролю цілісності і підтвердження достовірності електронних документів.

Основні принципи електронного документообігу:

- одноразова реєстрація документа , що дозволяє однозначно ідентифікувати документ в будь інсталяції даної системи.
- Можливість паралельного виконання операцій , що дозволяє скоротити час руху документів і підвищення оперативності їх виконання.
- Безперервність руху документа, що дозволяє ідентифікувати відповідального за виконання документа (завдання) в кожен момент часу життя документа (процесу).
- Єдина (або узгоджена розподілена) база документної інформації , що дозволяє виключити можливість дублювання документів;
- Ефективно організована система пошуку документа, що дозволяє знаходити документ, володіючи мінімальною інформацією про нього;
- розвинена система звітності по різних статусах і атрибутам документів, що дозволяє контролювати рух документів по процесах документообігу та приймати управлінські рішення, ґрунтуючись на даних зі звітів.

Підсистемні компоненти:

- система автоматизації архівного діловодства необхідна для архівного підрозділу підприємства . Допомагає формувати архівні справи, номенклатуру справ, вести облік, контролювати дати обов'язкового зберігання і т.д.
- електронний архів фінансової (платіжної) документації забезпечує централізований облік і зберігання електронних образів первинних фінансово -економічних документів, договорів та інших документів, що мають відношення до фінансово-економічної діяльності підприємства .
- електронний архів проектної документації дозволяє проектним організаціям зберігати весь спектр документів, як то: креслення, кошториси, пояснювальні записки та багато іншого.
- електронний архів нормативних документів являє собою організований каталог всіх стандартів підприємства, закуплених і (або) розроблених ним. Створення такого архіву тісно пов'язане з проблемою авторського права компаній-розробників стандартів і часто буває однією з умов проходження сертифікації.

- електронний архів технічної документації призначений для організацій, чия діяльність пов'язана з поставкою товарів і забезпечує зберігання супутньої технічної документації (керівництва користувача, технічні характеристики та інших).
- електронний архів юридичної документації , що містить договори та супровідну документацію;
- електронний архів кадрової документації , що використовується для зберігання особистих справ , трудових договорів , наказів і розпоряджень , інструкцій і регламентів всередині компанії.

–

1.2.4 Цілі і критерії створюваної автоматизованої інформаційної системи

Цілями створення автоматизованої системи управління документообігу ВП Південноукраїнська АЕС є:

- підвищення ефективності управління підприємством;
- оптимізація ведення фінансової документації;
- оптимізація обігу інформації;
- зменшення ризику помилок при прийнятті рішень через недостовірність інформації про функціонування підприємства;
- зменшення витрат на документообіг.

Критеріями для якісної оцінки проектованої системи вибираємо:

- надійність функціонування системи;
- можливість підтримки інформації в актуальному стані;
- швидкість обробки даних;
- простота використання системи;
- допустимі фінансові витрати на створення і функціонування системи.

1.2.5 Функції та задачі створюваної системи документообігу

Головним завданням створюваної автоматизованої системи є підвищення ефективності роботи підрозділів підприємства, але також система повинна вирішувати такі завдання:

- автоматизувати створення документації;
- автоматизувати пошук и знаходження документів;
- автоматизувати обмін документації між усіма департаментами.

Функції створюваної системи:

- ведення документів (створювання, видалення, редагування, отримання інформації, перегляд, передача документів);
- пошук
- пошук документів (за назвою, по виконавцю, за датою створення, по замовнику);
- пошук проектів (за назвою, по виконавцю, за датою створення, по замовнику);
- генерація звітів;
- формування списку проектів;
- формування списку робочої документації по кожному проекту;
- формування списку робочої документації по кожному виконавцю;
- адміністрування.

1.3 Разработка концепції інформаційної системи

1.3.1 Загальні вимоги до інформаційної системи

Виходячи з аналізу функцій систем і вимог до систем, можна визначити, що дані системи повинні включати основні компоненти:

- Система управління базами даних – ядро будь-якої інформаційної системи;

- Серверна платформа та програмне забезпечення - один з основних факторів ідеології організації інформаційної системи;
- Клієнтська платформа та клієнтське програмне забезпечення;
- Мережева середовище;
- Засоби розробки.

1.3.2 Опис результатів вивчення об'єктів

У результаті вивчення структури підприємства ВП «Південноукраїнська АЕС» та результатів функціонування його підрозділів були розглянуті і сформульовані проблеми, вирішення яких вимагає розробки автоматизованої інформаційної системи підприємства і дозволить подолати існуючі на підприємстві проблеми.

У результаті аналізу існуючої системи були відзначені істотні недоліки, що знижують ефективність підприємства, сформовано вимоги до системи, а також до виконуваних нею функцій і покладених на неї завдань. Система повинна забезпечити:

- управління документообігом;
- обмін документації між департаментами;
- пошук документів та проектів;
- генерація звітів.

Грунтуючись на результатах вивчення об'єкта автоматизації та з метою вибору оптимального рішення необхідно провести порівняльний аналіз можливих варіантів концепцій системи та оцінити переваги та недоліки кожної з них на підставі різних критеріїв і вимог користувача.

1.3.3 Варіант концепції інформаційної системи на базі FTP сервера

FTP розшифровується як «Протокол передачі даних» (File Transfer Protocol). Протоколом називається мова, на якому комп'ютери можуть розмовляти один з одним. Протокол FTP дозволяє вільно передавати файли і папки через Інтернет.

Електронний архів, система електронного архіва — система структурованого зберігання електронних документів, що забезпечує надійність зберігання, конфіденційність і розмежування прав доступу. Як видно з рисунка 1.5 система будувалася на базі FTP Сервера.

FTP дозволить підвищити продуктивність і підняти ефективність робочих процесів, значно знизити витрати та убезпечити документи. Помітно оптимізується командна робота на підприємстві. FTP сервер кардинально спрощує обмін файлами великого обсягу з партнерами підприємства через інтернет без використання електронної пошти.

Переваги: простота у використанні, дешево, відсутні витрати на навчання персоналу, забезпечує можливість пошуку, отримання та передачі файлів великого об'єму через інтернет з одного комп'ютера на інший, електронний архів конструкторської документації дозволяє організувати доступ до копій креслень, незалежно від їх давності.

Недоліки:

- даний варіант концепції не дозволяє розподілити ресурс між департаментами;
- відсутній модуль для службових повідомлень, який би дозволяв своєчасно персоналу дізнаватися про зміни в структурі проекту (додавання нового документа, створення нового проекту, зміни документації).
- відсутній модуль генерації звітів;
- відсутній модуль пошуку документів та проектів;

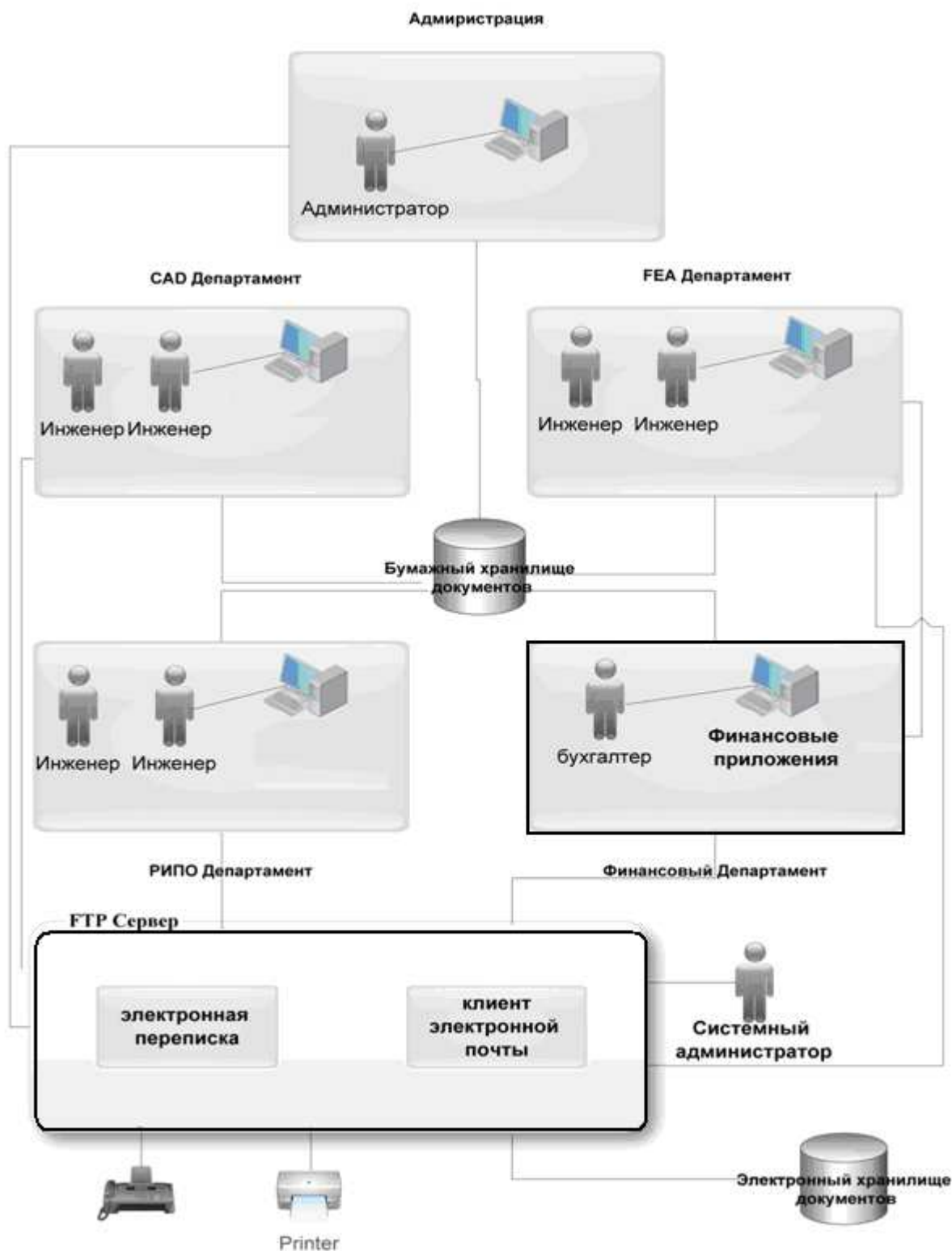


Рис. 1.5 – Варіант концепції інформаційної системи на основі бази FTP сервера

1.3.4 Варіант концепції «Система управління документами»

Система автоматизації документообігу, система управління документами (СУД) - автоматизована багатокористувачка система, що супроводжує процес управління роботою ієрархічної організації з метою забезпечення виконання цією організацією своїх функцій (рисунок 1.6). При цьому

передбачається, що процес управління спирається на людиночитасмі документи, що містять в слабо формалізованій формі інструкції для співробітників організації, необхідні до виконання.

Переваги:

- Адаптивність;
- Надійність;
- Захищенність.

Недоліки:

- необхідно навчити велика кількість персоналу пользоваться пропонованої інформаційною системою.

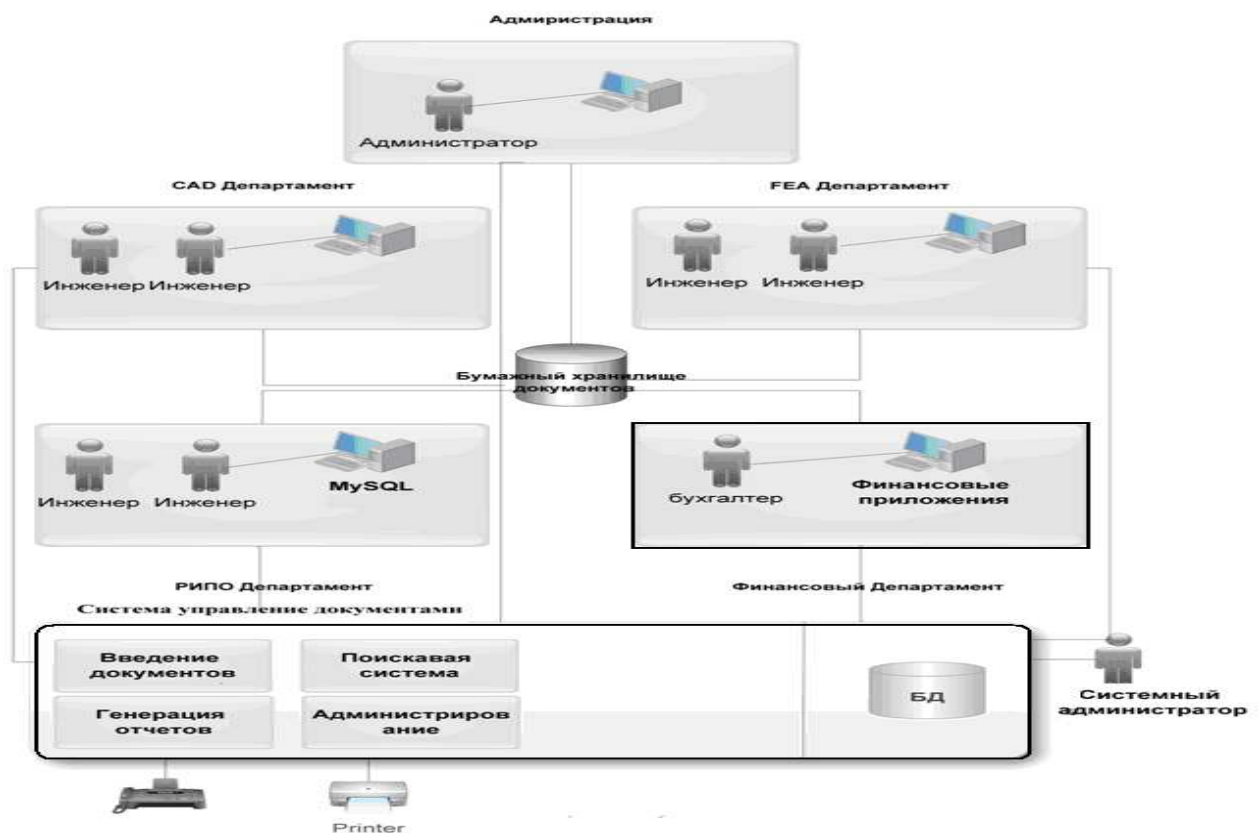


Рис. 1.6 – Вариант концепции «Система управления документами»

1.4 Функционально-вартісний аналіз

На основі вищенаведеного аналізу існуючих рішень проведемо функціонально-вартісний аналіз з метою визначення найбільш правильного і обгрунтованого впровадження системи.

Проведемо функціонально-вартісний аналіз двох варіантів впровадження існуючих концепції автоматизованої інформаційної системи.

- 1) Автоматизована система на базі FTP сервера;
- 2) Система управління документами (СУД);

На першому етапі функціонального аналізу необхідно визначити ступінь відповідності кожного варіанта концепції системи функцій, які повинна виконувати система. У Таблиці 1.2 наведено експертні оцінки за обраними варіантами по кожному з параметрів. Оцінки отримані в результаті співбесіди з кількома експертами, усереднені і округлені.

Таблиця 1.2 – Основні параметри системи

Параметр, F_i	Оцінка в балах за варіантом №, V_i	
	1	2
Продуктивність	4	9
Надійність	5	7
Захищеність	7	7
Адаптивність	3	8
Документальність	5	6
Всього	24	37

На другому етапі функціонального аналізу необхідно визначити вагові коефіцієнти для кожного з параметрів. У даному випадку вони також вибираються експертним шляхом. Значення вагових коефіцієнтів для кожного з параметрів наведені в Таблиці 1.3.

Таблиця 1.3 – Вагові коефіцієнти для параметрів системи

Параметр, F_i	Коефіцієнт, ϕ_i
Продуктивність	0.30
Надійність	0.20
Захищеність	0.30
Адаптивність	0.15
Документальність	0.05

Після визначення вагових коефіцієнтів можна визначити коефіцієнти технічного рівня для кожного з параметрів по кожному варіанту за формулою:

$$КТУ[F_i] = V_i \phi_i \quad (1.1)$$

За даними Таблиці 1.2 розраховується коефіцієнт технічного рівня КТР для кожного з варіантів за формулою:

$$КТР[F_i] = \sum_i КТР[F_i] \quad (1.2)$$

Результати розрахунків приведені в Таблиці 1.4

Таблиця 1.4 – Результати розрахунків

Параметр, F_i	Варіант	Оцінка параметра, V_i	Ваговий коефіцієнт, ϕ_i	Показник технічного рівня $КТР[F_i]$
Продуктивність	1	4	0.30	1.2
	2	9		2.7
Надійність	1	5	0.20	1.0
	2	7		1.4
Захищеність	1	7	0.30	2.1
	2	7		2.1
Адаптивність	1	3	0.15	0.45
	2	8		1.20
Документальність	1	5	0.05	0.25
	2	6		0.30

Таблиця 1.5– Коефіцієнти технічного рівня

Варіант	Коефіцієнт технічного рівня (КТР)
2	7.7
3	6.8

Кращим на етапі функціонального аналізу є варіант, що має максимальне значення коефіцієнта технічного рівня КТР, тобто варіант концепції «СУД».

Аналіз функцій, реалізуються системою, повинен бути доповнений вартісним аналізом. Для цього розглядаються зведені витрати Z_j по кожному варіанту:

$$Z_j = C_j + E_n \times K_j, \quad (1.3)$$

де C_j - поточні витрати по j -му варіанту; K_j - капітальні вкладення по j -му варіанту;

E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень, що дорівнює 0,15.

Вільні витрати Z_j розраховують у сфері виробництв (у розробника) або у сфері експлуатації (у користувача). У нашому випадку доцільно розраховувати витрати в сфері виробництва. Поточні витрати при розробці C_j - це собівартість створення системи C_c :

$$C_c = C_z + C_v + C_m + C_n, \quad (1.4)$$

де C_z – заробітна плата розробників;

C_v – відрахування на соціальні заходи (47,5% від фонду оплати праці). C_m - вартість машинного часу, необхідного для розробки та налаштування системи;

C_n – накладні витрати в розмірі 50-100% від витрат на оплату праці (містять оплату праці управлінського персоналу з нарахуваннями оплат службових відряджень; оплату консультативно-інформаційних послуг оплату ремонту і техобслуговування основних фондів, крім ПК; оплату оренди приміщення).

Капітальні вкладення K_j - це витрати, які мають разовий характер, в даному випадку внесемо сюди вартість покупного ліцензійного програмного забезпечення, технічного та організаційного забезпечення, необхідного для створення системи. Необхідного програмного забезпечення на сьогоднішній день наведена в Таблиці 1.6.

Таблиця 1.6 – Вартість програмного забезпечення

Варіант	Найменування ПЗ	Вартість, грн.	Кіл-сть	Разом, грн.
1	Linux / Ubuntu 10.10 Server	0	1	0
2	Linux / Ubuntu 10.10 Server	0	1	0
	Eclipse	0	2	0
	Python 2.7 / Django 1.2.3	0	1	0
	MySQL Database 5.1	0	1	0
3	Linux / Ubuntu 10.10 Server	0	0	0

За результатами функціонального і вартісного аналізу можна розрахувати коефіцієнт техніко-економічного рівня по кожному з варіантів за формулою:

$$K_{TEVj} = \frac{K_{TVj}}{3_j} \quad (1.6)$$

Результати розрахунку приведені в Таблиці 1.7

Таблиця 1.7 – Коефіцієнти техніко-економічного рівня

Варіант	КТЕУj
1	0.00103
2	0.00105

Максимальне значення коефіцієнта КТЕУj має другий варіант концепції, тобто «Автоматизована система на базі системи управління документами».

1.5 Оцінка надійності

Серверне обладнання, найчастіше, призначене для забезпечення роботи сервісів в режимі 24 / 7, тому комплектується надійними елементами, що дозволяють забезпечити "п'ять дев'яток" (99,999 % доступність сервера або простий даної системи складає менше 6 хвилин на рік).

Для проектування АІС нам потрібен веб сервер, сервер додатків, СУБД, дисковий сховище і доступ в Інтернет. Для простоти будемо вважати, що програмне забезпечення «сервер додатків» включає в себе веб сервер і буде

встановлено на одному апаратній сервері , СУБД - Reliability - шукану величину доведеться самостійно оцінити і призначити . Кожен компонент є єдиною точкою відмови , тому на робочій схемі для розрахунку надійності вони з'єднані послідовно . Зауважимо , що це не схема з'єднання компонентів інфраструктури , а лише схема розрахунку надійності повинна бути отримана від постачальника компонента (обладнання, програмного забезпечення або послуги) . Якщо це, з якихось причин неможливо (наприклад , для програмних компонентів значення надійності , як правило , невідомо) - далі будемо користуватися терміном «надійність» - будемо проект інфраструктури. Під кожним компонентом напишемо параметри його доступності. Доступність кожного компонента , отже , розраховуємо надійність. Оскільки нас послідовне з'єднання компонентів, то величини надійності перемножуються:

$$\text{Reliability} = (0,99999 \times 0,99999 \times 0,985 \times 0.98) \times 100\% = 96,528\%.$$

1.6 Постановка задачі

Розробити проект автоматизованої інформаційної системи, на основі варіанту концепції «Система управління документами». Ця система володіє наступними можливостями:

- вести документ (створити, видалити, редагувати, отримати інформацію, створити нову версію документа);
- пошук документів та проектів (пошук за назвою, по виконавцю,, по замовнику, за датою створення);
- генерація звітів;
- адміністрування системи управління документами.

1.7 Умови приймання системи

Приймання створеної системи здійснюється представниками виконавця і замовника відповідно до технічного завдання. Складається двосторонній акт, що стає підставою для розрахунків.

2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМІ ДОКУМЕНТООБІГУ ВП «ПІВДЕННОУКРАЇНСЬКА АЕС»

2.1 Загально-системні проектні рішення

2.1.1 Організаційна і функціональна структура підприємства

У процесі свого функціонування система взаємодіє з користувачами з різних підрозділів для виконання ними різних функцій. Схема цієї взаємодії, представлена в вигляді діаграми варіантів використання.

Діаграму використання можна представити як деякий процес покрокового спуску від найбільш загальної і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи. Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання, яка описує функціональне призначення системи або, іншими словами, те, що система робитиме в процесі свого функціонування. Діаграма варіантів використання є вихідним представленням концептуальної моделі системи в процесі її проектування і розробки.

Розробка діаграми варіантів використання переслідує цілі:

- Визначити загальні кордони і контекст модельованої наочної області на початкових етапах проектування системи;
- Сформулювати загальні вимоги до функціональної поведінки проектованої системи;
- Розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;
- Підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

Діаграма, яка буде задовольняти вимогам дипломного проекту представлена на Рисунку 2.1 нижче.

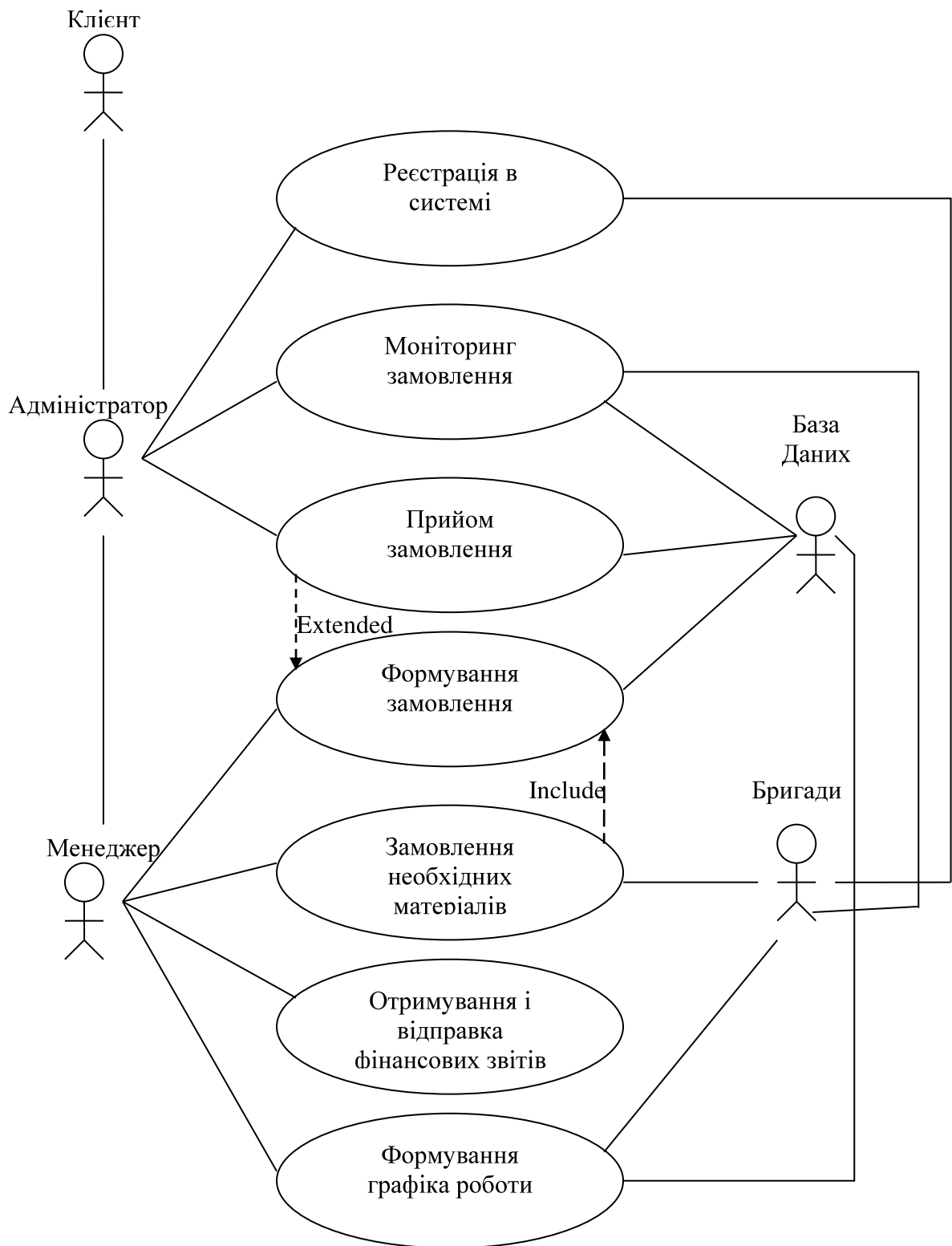


Рисунок 2.1 – Діаграма використання

На етапі аналізу вимог і визначення специфікацій діаграма переходів станів демонструє поведінку програмної системи, що розробляється, при

здобутті дій, що управляють. Під діями, що управляють, або сигналами в даному випадку розуміють інформацію, що управляє, отримується системою ззовні. Наприклад, діями, що управляють, вважають команди. Отримавши таку дію, що управляє, система, що розробляється, повинна виконати певні дії або залишитися в тому ж стані, або перейти в інший стан взаємодії із зовнішнім середовищем.

Розглянемо діаграму переходів станів представлену на Рисунку 2.2

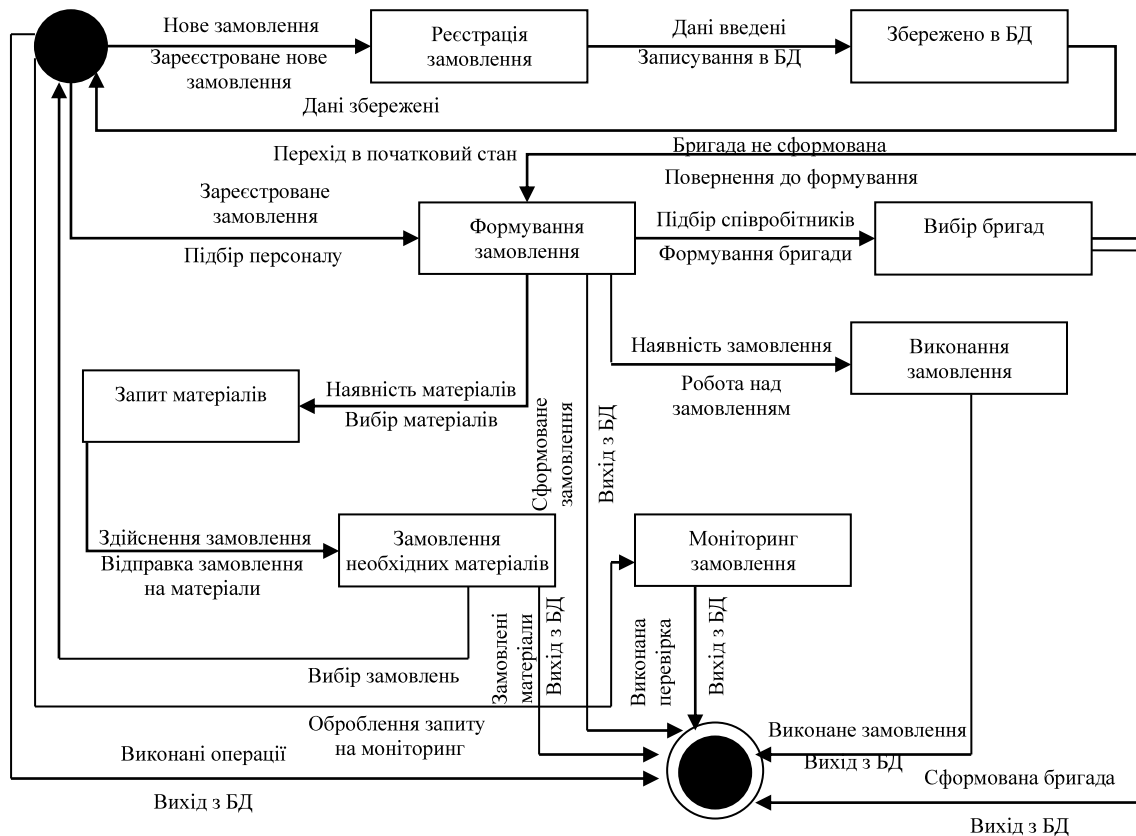


Рисунок 2.2 – діаграма переходів станів

Коли замовник приходить на підприємство адміністратор повинен його зареєструвати в системі. Після того як замовник зареєстрований всі дані про його замовлення записуються в базу даних. Після чого адміністратор повинен сформувати замовлення на послуги. Для цього йому потрібно вибрати необхідні для робіт матеріали, які є на складі або ж замовити у постачальника, сформувати бригаду працівників, які будуть виконувати замовлення (для цього необхідно знати графік роботи кожного співробітника). Після того як замовлення

сформоване стан його зберігається в базі, тобто етап виконання замовлення. Кінцевою функцією є моніторинг замовлення. На цьому етапі відбувається безперервне стеження за процесом з метою виявлення кількості заявок.

2.1.2 Загальний опис інформаційної системи ОП «Південно-українська АЕС»

Інформаційна система документообігу замовлень представляє собою комплексний програмно-апаратний комплекс контролю й управління інформацією, призначений для контролю і управління інформацією на робочих місцях співробітників та серверній мережі.

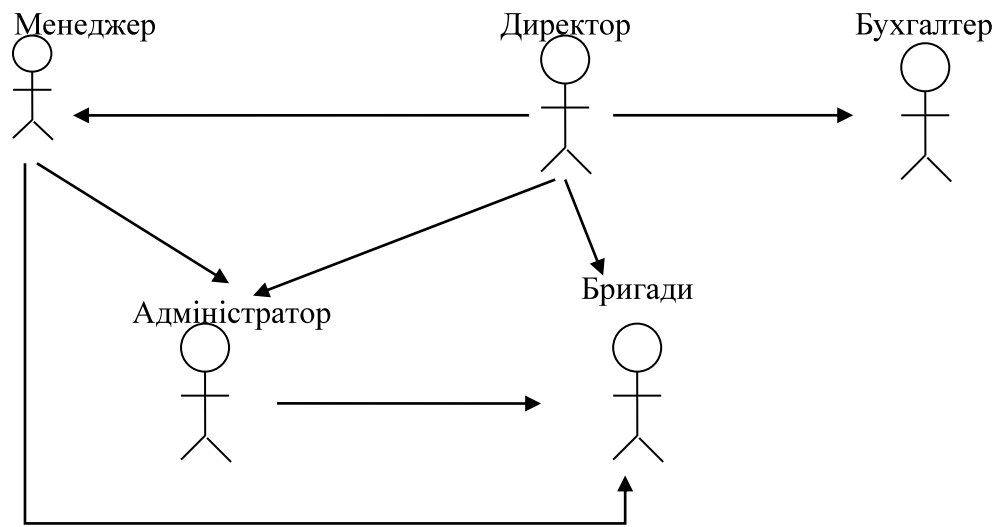
Дана система управління документообігом виконує наступні функції управління:

- управління доступом: привілеями та паролями;
- управління прийомом замовлень;
- управління формуванням замовлень;
- управління замовленнями матеріалів;
- управління формуванням графіків роботи;
- управління моніторингом замовлень;
- управління отриманням та відправкою звітів.

2.2 Рішення по організаційному забезпеченню

2.2.1 Організаційна структура

У ході впровадження і використання інформаційної системи, її функціями охоплюються всі суб'єкти підприємства. Для кожного з департаментів організаційна структура має приблизно однаковий вигляд, за винятком тих, де роль менеджера зазвичай відіграє головний інженер. Схема організаційної структури підприємства приведе на рисунку 2.3.



Ри
су
но
к
2.3
—
Су
б'є
кт

и підприємства, які взаємодіють

2.2.2 Опис організації структури підприємства

Директор:

- слідує за роботою всього колективу;
- керує та контролює фінансові і робочі процеси підприємства;
- видає накази на роботу з клієнтами.

Бухгалтер:

- веде облік і контролює фінансові операції на підприємстві;
- своєчасно звітує у державних інспекціях;
- надає директору звіт про фінансовий стан підприємства і складає економічний прогноз роботи підприємства;
- нараховує заробітну плату персоналу.

Менеджер:

- займається безпосередньо роботою з клієнтами і підготовкою документації для формування замовлення;
- робить розрахунок кошторисної вартості і формування рахунка з указівкою суми і реквізитами замовника;
- по виконанню замовлення менеджер і замовник підписують акт виконаних робіт, що засвідчує факт прийому замовником замовлення без зауважень і претензій;

- виконує функції по інформаційному забезпеченню.

Адміністратор:

- відповідальний за стан обладнання і мережі;
- розраховує відрядні для бригад;
- частково виконує функції менеджера (проводить моніторинг звітів, реєструє звіти, формує звітну статистику).

Бригади. До складу бригади входять робітники, які повинні виконувати наступні дії:

- виконують доручення адміністратора;
- обробляють замовлення;
- надають замовлення в поставлений термін замовнику;
- подають заявки на матеріали.

2.2.3 Організаційно-кадрові заходи

При впровадженні інформаційної системи на підприємстві до організаційної структури підприємства необхідно включити віддаленого висококваліфікованого системного адміністратора, який буде виконувати функції по усуненню технічних недоліків в роботі. Всі дії які полягають по інформаційному забезпеченню ляжуть на плечі менеджера та частково адміністратора.

2.3 Робочий проект

2.3.1 Рішення з інформаційного забезпечення

2.3.2 Опис інформаційного забезпечення

У діловій сфері часто приходится працювати з даними з різних джерел, кожне з яких пов'язане з певним видом діяльності. Для координації всіх цих даних необхідні певні знання й організаційні навички.

Електронною базою даних (БД) називається послідовність даних заданої структури, записана на магнітний диск комп'ютера.

Системи управління базами даних (СУБД) є набором програмних засобів, необхідних для створення, використання і підтримки баз даних.

База даних – це набір даних з наступними властивостями:

- дані логічно пов'язані між собою і несуть відповідну інформацію;
- структура баз даних звичайно відповідає тому специфічному набору даних, які вона містить;
- бази даних відображають тільки окремі аспекти реального світу, що дає змогу визначити їх як "мікросвіт".

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно оновлювати дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

Існує два типи систем управління БД: файлами і реляційними базами даних. В системах управління файлами (File management systems), іноді їх ще називають плоскими файловими базами даних, дані вводяться і зберігаються без індексування, а при пошуку і створенні вихідних форм обробляються послідовно. Тому ці системи недостатньо гнучкі при маніпулюванні даними.

Системи управління реляційними базами даних (Relational database management systems) дозволяють користувачам маніпулювати даними більш вільно завдяки збереженню всіх даних тільки один раз і опису зв'язків між ними. Зв'язки являють собою загальні елементи даних, таких як прізвище покупця або номер його чекової книжки. Дані кожного набору інформації можна знайти, відобразити і відновити на підставі даних іншого набору.\

2.3.3 Опис організації інформаційної бази даних

При проектуванні структури нової бази даних визначають суть предметної області, яка повинна знайти своє віддзеркалення в базі даних.

Аналіз предметної області зазвичай здійснюється:

- на підставі існуючих відомостей про предметної області, в масштабах в яких вона має бути представлена в створюваній базі даних і працюючих з нею додатках;
- виходячи з цілей проектування програмного середовища;
- на підставі уявлення про те, яке місце база даних і застосування, що працюють з нею, займають в структурі експлуатуючої її організації.

Зрештою аналіз предметної області повинен привести до створення проекту бази даних.

Спочатку необхідно змалювати суть і зв'язки між ними. Як правило, кожній суті (об'єкту) в базі даних відповідає таблиця. Потім для кожної таблиці бази даних приводиться список полів запису.

Початкова стадія проектування включає аналіз об'єктів реалних подій, які будуть відбиті в базі даних.

Формування концептуальної моделі бази даних включає:

- ідентифікацію функціональної діяльності предметної області;
- ідентифікацію об'єктів (суті), які здійснюють цю функціональну діяльність, і формування з їх операцій послідовності подій, які допоможуть ідентифікувати всю суть і взаємозв'язки між ними;
- ідентифікацію характеристик цієї суті;

- ідентифікацію взаємозв'язків між суттю;
- функціональну діяльність і формування з їх операцій.

На підприємство звертається багато клієнтів тож для обліку їх ведеться певна анкета. Замовлення яке зіставляється менеджером теж заповнюється на певному прикладі анкети.

Після того як адміністратор зареєстрував клієнта в базі даних він здійснює пошук співробітників для виконання замовлення клієнта.

З вище сказаного в якості функціональної діяльності визначається необхідність видачі інформації про клієнтів фірми, складання заявок на покупку матеріалів для виконання замовлення. Формування звітів, які відображають результати роботи фірми.

В створеній базі даних існує 6 об'єкта:

1. Клієнт
2. Замовлення
3. Постачальники
4. Співробітники
5. Бригада
6. Матеріали

До об'єктів моделі вибираються основні атрибути.

Основні атрибути для Клієнта:

- ФІО – прізвище, ім'я та по батькові клієнта;
- Код – код клієнта при реєстрації;
- Адреса – адреса клієнта, для встановлення зв'язку між клієнтом та виконавцями замовлення;
- Телефон – номер телефону клієнта;
- Вид замовлення – різновид замовлень, побажання клієнта по замовленню;
- Місце Виконання – адреса по якій буде здійснено замовлення клієнта;
- Дата Виконання – термін, в який потрібно здійснити замовлення клієнта;

- Примітки – примічання для зручності адміністратору.

Основні атрибути для Замовлення:

- Код замовлення – код замовлення при реєстрації замовлення;
- Код – код клієнта при реєстрації;
- Дата Виконання – термін, в який потрібно здійснити замовлення клієнта;
- Код бригади – персональний код співробітників, які входять до бригади;
- Ціна замовлення – встановлена грошова винагорода за здійснене замовлення.
- Вид замовлення – різновид замовлень, побажання клієнта по замовленню.

Основні атрибути для Постачальника:

- Код постачальник – персональний код постачальника;
- Назва постачальника – назва фірми, яка здійснює поставку матеріалів;
- Адреса – адреса постачальника, для здійснення зв'язку між постачальником та співробітниками фірми «Пестиж Центр»;
- Телефон – телефон постачальників;
- Факс – факс постачальника;
- Умови оплати – умови при яких здійснюється оплата матеріалів;
- Код Матеріалу - персональний код кожного матеріалу.

Основні атрибути для Співробітників:

- ФІО – прізвище, ім'я та по батькові співробітників фірми;
- Код співробітника – персональний код кожного робітника фірми;
- Адреса – адреса співробітника;
- Телефон Домашній – номер телефону робітника фірми;
- Посада – посада, яку займає працівник на підприємстві;
- Телефон Робочий – номер телефону робітника фірми;
- Код бригади - персональний код співробітників, які входять до бригади.

Основні атрибути для Бригади:

- Код бригади - персональний код співробітників, які входять до бригади;
- Код співробітника - персональний код кожного робітника фірми;
- ФІО – прізвище, ім'я та по батькові співробітників фірми;
- Опис – опис самого матеріалу;
- Посада – посада, яку займає працівник на підприємстві.

Основні атрибути для Матеріалів:

- Код - персональний код кожного матеріалу;
- Марка – назва чи персональний код;
- Код постачальник – персональний код постачальника;
- Ціна – ціна за одиницю матеріалу;
- Опис – опис самого матеріалу;
- На складі – товар який присутній в даний момент;
- Заказано- товар, який заказаний;
- Код Замовлення - код замовлення при реєстрації замовлення

Після вибору атрибутів визначаються зв'язки між об'єктами. Залежно від умов, зв'язки між об'єктами можуть бути: один до одного, один до багатьом і багато до багатьом. В результаті була спроектована концептуальна модель, яка зображена на Рисунку 2.4.

На основі даних будуємо концептуальну модель:

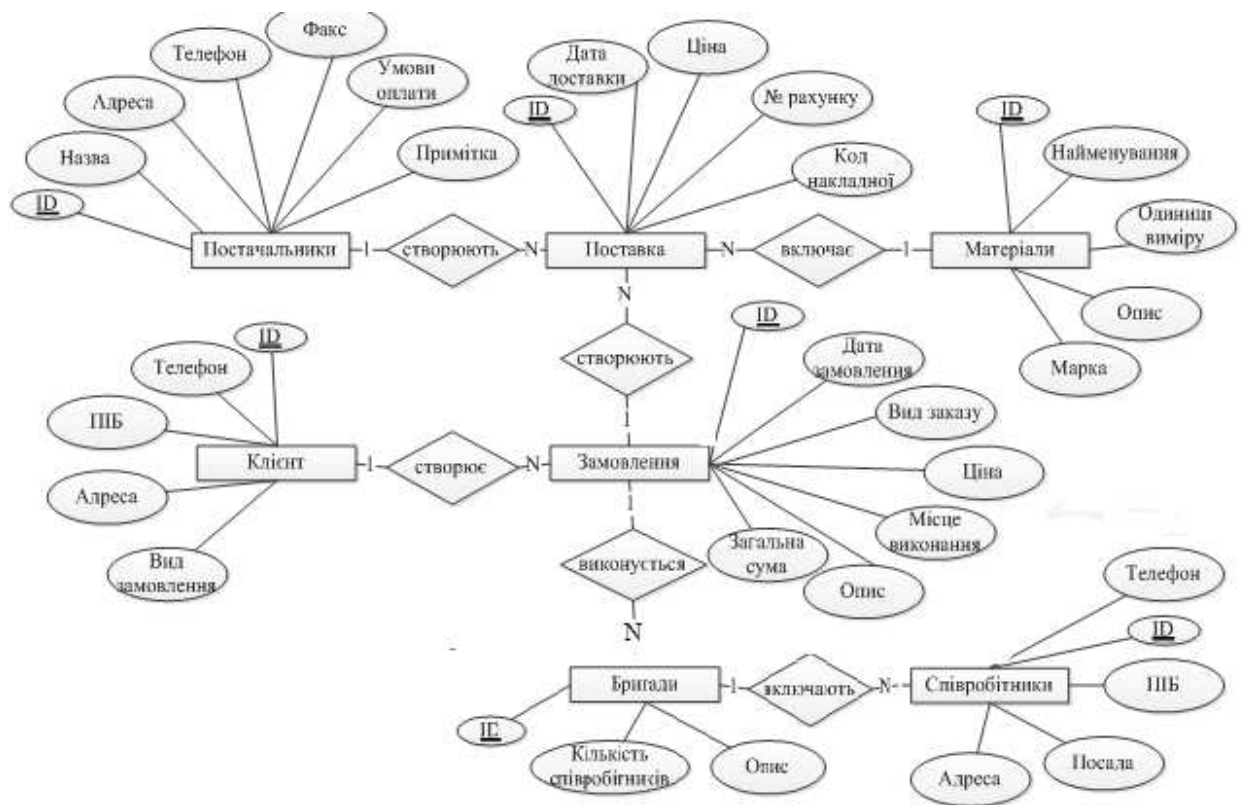


Рисунок 2.4 – концептуальна модель

2.3.4 Логічна модель даних

На підставі розробленої вище концептуальної моделі (рисунок 2.4) побудуємо логічну модель даних.

Логічна структура реляційної бази даних в Access є адекватним відображенням отриманої інформаційно-логічної моделі, що не вимагає додаткових перетворень. Кожен інформаційний об'єкт моделі даних відображується відповідною реляційною таблицею. Структура реляційної таблиці визначається реквізитним складом відповідного інформаційного об'єкту, де кожен стовпець (поле) відповідає одному з реквізитів об'єкту. Ключові реквізити об'єкту утворюють унікальний ключ реляційної таблиці. Для кожного стовпця задається формат і розмір даних. Рядки (записи) таблиці відповідають екземплярам об'єкту і формуються при завантаженні таблиці.

Зв'язки між об'єктами моделі даних реалізуються однаковими реквізитами – ключами зв'язку у відповідних таблицях. При цьому ключем зв'язку завжди є унікальний ключ головної таблиці. Ключем зв'язку в підлеглий таблиці є або

деяка частина унікального ключа в ній, або поле, що не входить до складу первинного ключа (наприклад, код Клієнта в таблиці КЛІЄНТ). Ключ зв'язку в підлеглий таблиці називається зовнішнім ключем.

На основі концептуальної моделі будується реляційна модель, тобто для кожного об'єкту створюється таблиця, що містить всі атрибути даного об'єкту. Унікальний атрибут буде первинним ключем. Для побудови зв'язків між таблицями додаються поля, які будуть зовнішніми ключами. В результаті, побудувавши зв'язки за допомогою первинних і внутрішніх ключів, отримуємо реляційну модель. Так як дана концептуальна модель містить декілька зв'язків багато-до-багатьох ми повинні нормалізувати ці зв'язки, кожен його неключовий атрибут повинен залежати від первинного ключа.

Нормалізація – це процес, що дозволяє гарантувати, ефективність структур даних в реляційній базі даних. Реляційна база даних вважається ефективною, якщо вона володіє наступними характеристиками: відсутність надмірності, мінімальне використання null-значень, запобігання втраті інформації.

На рисунку 2.5 зображено логічну модель даних.

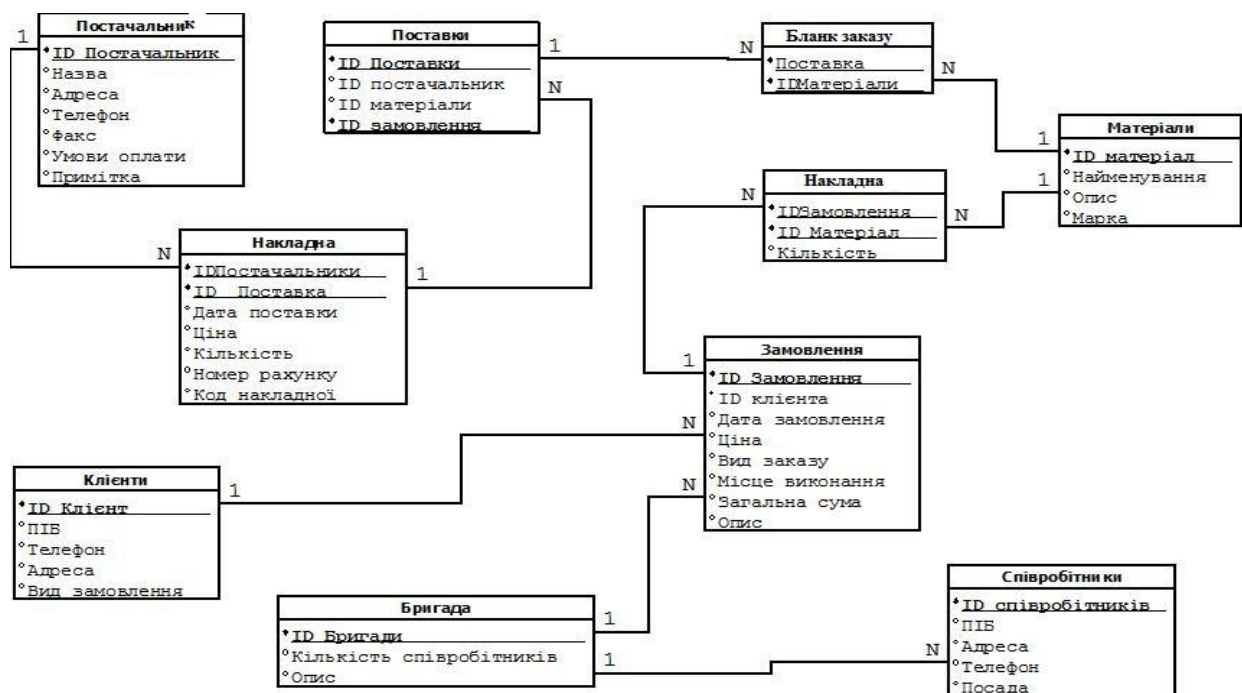


Рисунок 2.5 – Логічна модель

2.3.5 Фізична модель даних

Фізична модель БД визначає спосіб розміщення даних на носіях (пристроях зовнішньої пам'яті), а також спосіб і засоби організації ефективного доступу до них. Оскільки СУБД функціонує в складі і під управлінням операційної системи, то організація зберігання даних і доступу до них залежить від принципів і методів управління даними операційної системи.

До питань організації даних відносяться:

- вибір типу запису – одиниці обміну в операціях введення-виводу;
- вибір способу розміщення записів у файлі і, можливо, методу оптимізації розміщення;
- вибір способу адресації і методу доступу до записів.

Стадія фізичного проектування БД в загальному випадку включає:

- вибір способу організації БД;
- розробку специфікації внутрішньої схеми;
- опис відображення концептуальної схеми у внутрішню. На відміну від ранніх СУБД, багато сучасних систем не надають розробникові якого-небудь вибору на цій стадії. Реально до питань проектування фізичної моделі можна віднести:
- вибір схеми розміщення даних (розділення по файлах або тип `raid`-масиву);
- визначення числа і типа індексів (наприклад, кластеризований або некластеризований в разі MS SQL Server).

Спосіб зберігання БД визначається механізмами СУБД автоматично за умовчанням на основі специфікацій концептуальної схеми БД, і внутрішня схема в явному вигляді в таких системах не використовується. Зовнішні схеми БД зазвичай конструюються на стадії розробки додатків. Фізична модель даних представлена на рисунку 2.6, та описана в Таблиці 2.1.

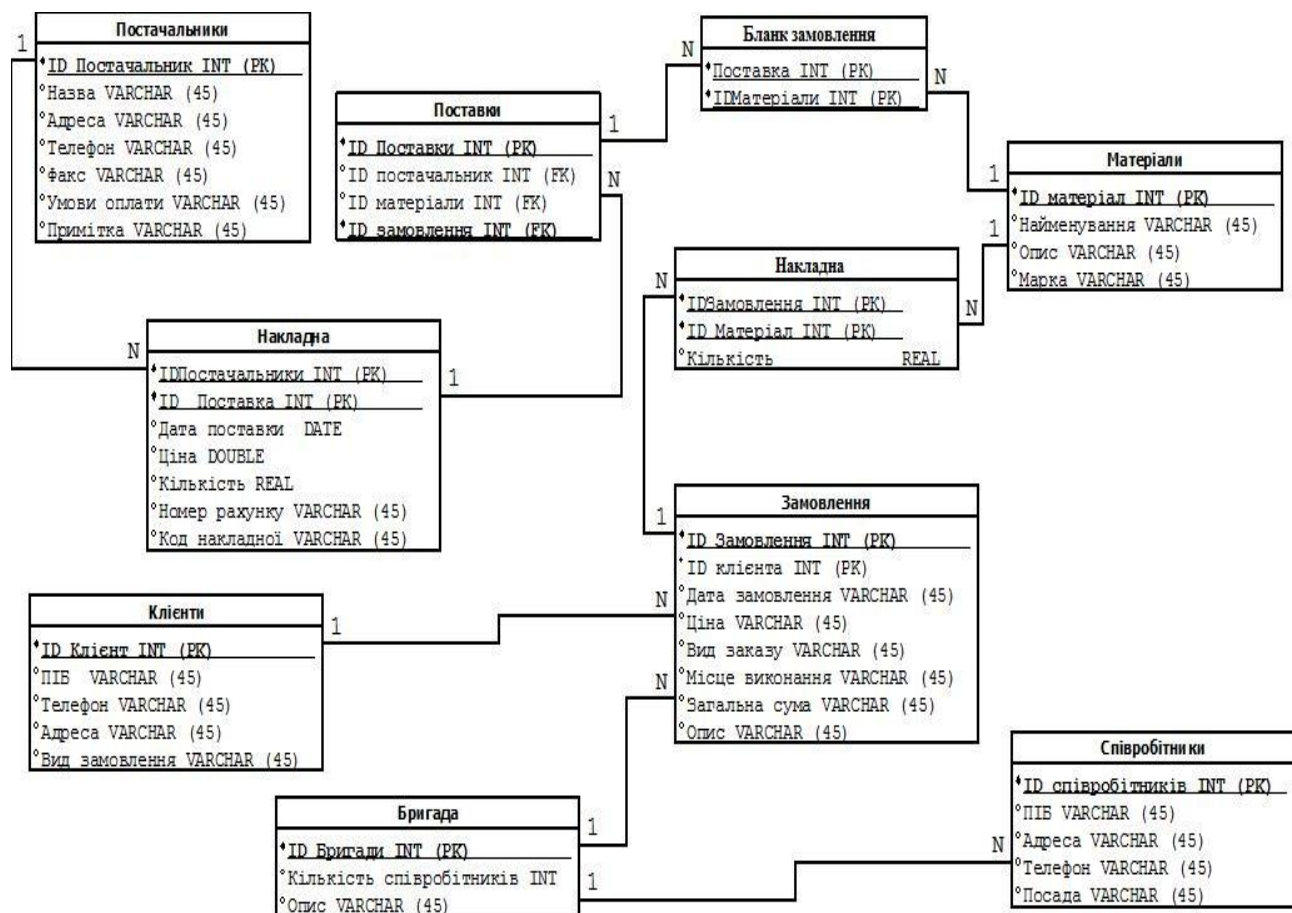


Рисунок 2.6 – Фізична модель

Таблиця 2.1 Фізична модель даних

Назва об'єкту (Назва таблиці бази даних)	Позначення об'єкту	Властивості (Опис поля)	Межі (Тип поля)
Постачальник	ID Постачальника	Код Постачальника (ключ)	Лічильник
	Назва Постачальника	Назва Постачальника	Текстовий (45 символів)
	Адреса Постачальника	Місто, Вулиця	Текстовий (45 символів)
	Телефон	Номер Телефона	Текстовий (45 символів)
	Умови оплати	Умови оплати	Текстовий (45 символів)
	Факс	Факс	Текстовий

Назва об'єкту (Назва таблиці бази даних)	Позначення об'єкту	Властивості (Опис поля)	Межі (Тип поля)
			(45 символів)
	Примітка	Примічання	Текстовий (45 символів)
Клієнт	Код Клієнта	Код Клієнта (ключ)	Лічильник
	ФІО	ФІО Клієнта	Текстовий (50 символів)
	Адреса	Адреса Клієнта	Текстовий (40 символів)
	Телефон	Телефон Клієнта	Текстовий (30 символів)
	Вид замовлення	Вид замовлення	Текстовий (20 символів)
Співробітники	Код Співробітника	Код Співробітника (ключ)	Лічильник
	ПІБ	ПІБ Співробітника	Текстовий (45 символів)
	Посада	Посада Співробітника	Текстовий (45 символів)
	Адреса	Адреса Співробітника	Текстовий (45 символів)
	Телефон	Телефон Співробітника	Текстовий (45 символів)
	Номер ідентифікаційного коду	Номер ідентифікаційного коду співробітника	Текстовий (45 символів)
Замовлення	Код замовлення	Код замовлення (ключ)	Лічильник

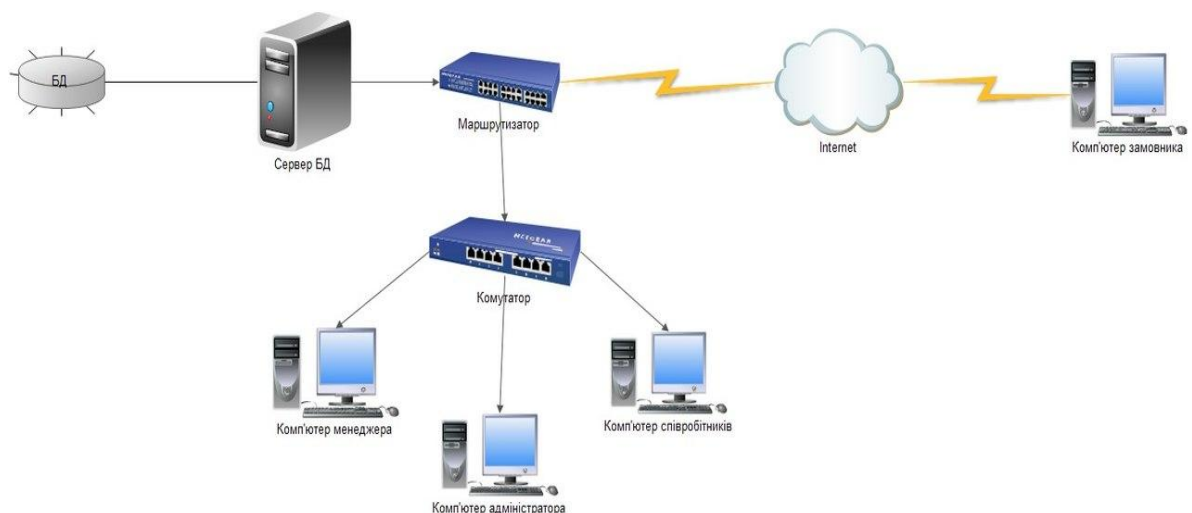
Назва об'єкту (Назва таблиці бази даних)	Позначення об'єкту	Властивості (Опис поля)	Межі (Тип поля)
	Код Замовника	Код Замовника	Числовий (довге ціле)
	Вид замовлення	Вид замовлення	Текстовий (20 символів)
	Ціна замовлення	Ціна замовлення	Грошовий
	Місце виконання	Місце виконання	Текстовий (45 символів)
	Опис	Опис	Поле MEMO

2.4 Рішення з технічного забезпечення

На рисунку 2.7 зображено схема комп'ютерної мережі на ВП «Південноукраїнська АЕС». Роз'яснення щодо елементів архітектури наводяться нижче.

Сервер – Файловий сервер підприємства на якому зберігаються документи та файли, що містять комерційну таємницю та сервер баз даних. Сервер БД обслуговує базу даних і відповідає за цілісність і збереження даних, а також забезпечує операції введення-виводу при доступі клієнта до інформації.

Комп'ютер менеджера, адміністратора, замовників та співробітників – комп'ютери через, які кожен з вище перелічених осіб має змогу отримати, занести чи змінити дані.



DI-824VUP+ - це безпроводний 802.11g маршрутизатор, який об'єднує функції широкого доступу до Інтернету з надійною захищеністю міжсітьовим екраном, вбудованим принт-сервером і 4-х портовим комутатором для підключення принтеру і робочих станцій. Він здійснює високу швидкість передачі по безпроводній мережі. Цей пристрій надає економічний спосіб встановлення безпечної і швидкодіючої мережі з каналом зв'язку без тісних місць до зовнішнього світу.

Fast Ethernet (Швидкий Ethernet) — термін, що описує набір стандартів Ethernet для пакетної передачі даних з номінальною швидкістю 100 Мбіт/с, що в 10 разів швидше за початкову для Ethernet швидкість у 10 Мбіт/с. Він визначений 1995 року в документі IEEE 802.3u. На сьогодні існують швидші в 10 (Gigabit Ethernet) і 100 (10 Gigabit Ethernet) разів стандарти технології Ethernet.

Всі відмінності технології Fast Ethernet від Ethernet зосереджені на фізичному рівні. Рівні MAC і LLC у Fast Ethernet залишилися абсолютно тими ж, і їх описують колишні глави стандартів 802.3 і 802.2. Тому, розглядаючи технологію Fast Ethernet, тут описуються тільки кілька варіантів її фізичного рівня.

Більш складна структура фізичного рівня технології Fast Ethernet викликана тим, що в ній використовуються три варіанти кабельних систем:

- волоконно-оптичний багатомодовий кабель, використовуються два волокна;
- кручена пари категорії 5, використовуються дві пари;
- кручена пари категорії 3, використовуються чотири пари.

2.5 Розробка рішення з програмного забезпечення

Основним засобом розробки програмного забезпечення системи є програмне середовище розробки Visual Studio. Мовою програмування для

розробки обрано мову PHP. Інструментом для ведення бази даних обрано СКБД MySQL.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом.

2.5.1 Розробка діаграми класів програмного забезпечення

На рисунку 2.8 представлена діаграма класів програмного забезпечення створення та зберігання звітної документації.

Для більш детального сприйняття ієрархії класів приведемо специфікацію головних класів:

1)Main – головний клас програми, він об'єднує в собі усі складові частини програмного забезпечення, а також відповідає за перевірку коректності проекту та його друк;

2)MainView – клас, що забезпечує відображення інтерфейсу програми, а також відповідає за завантаження та запис файлів проекту та встановлення опцій;

6) Page – клас, що є представленням конкретної сторінки в проекті, містить в собі різноманітні фігури, до яких можна отримати доступ за їх іменем (через метод `getFigureByName`);

7) AbstractFigure – клас, що є базовим для класів фігур графічного відображення, дозволяє намалювати фігуру (метод `draw`), дізнатися її ім'я (метод `getName`) та сторінку, до якої фігура належить (метод `getParentPage`).

2.5.2 Розгортання програмного забезпечення

Діаграма розгортання необхідна для візуалізації елементів та компонентів системи. Розробка діаграми розгортання являється останнім етапом специфікації моделі системи. При розробці даної діаграми визначається розподіл компонентів системи по її фізичним вузлам.

Серверна машина включає в собі БД та сервер БД, яка за допомогою вузлів з'єднується робочими станціями менеджера (включає його АРМ) та адміністратора (включає АРМ адміністратора). А також Серверна машина через вузол взаємодіє з комп'ютером співробітника, який включає його робоче місце.

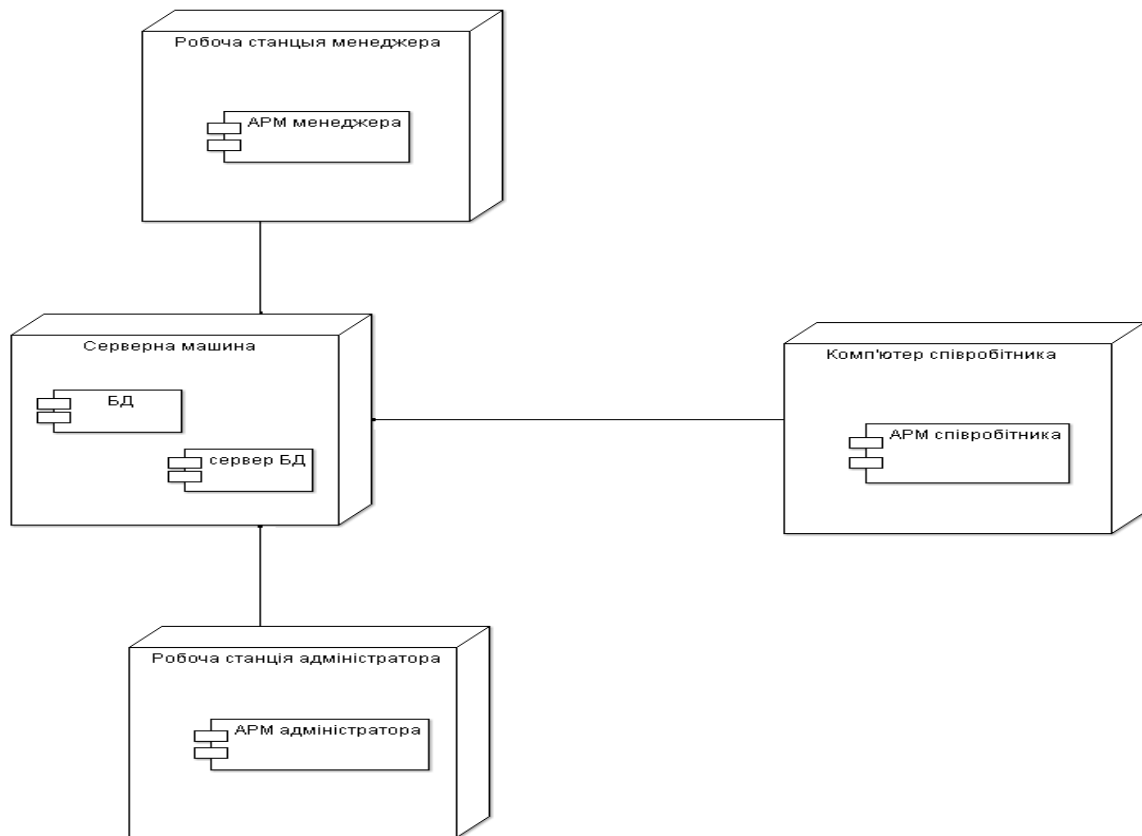


Рисунок 2.9 – Діаграма розгортання

3 РОЗРОБКА РОБОЧОЇ ДОКУМЕНТАЦІЇ СИСТЕМИ УПРАВЛІННЯ ДОКУМЕНТООБІГУ ВП ПУАЕС

При роботі з програмним забезпечення розробленої системи документообігу слід користуватися документами «Технічне завдання» (Додаток А), «Керівництво користувача» (Додаток Б), «Програма і методика випробувань» (Додаток В).

4 ОХОРОНА ПРАЦІ

4.1. Вступ

Охорона праці – система законодавчих актів, соціально-економічних, організаційних, технічних, гігієнічних, лікувально-профілактичних заходів і засобів забезпечення безпеки, збереження здоров'я і працездатності людини в процесі праці.

Як відомо, цілком безпечних і нешкідливих виробництв не існує, тому основною задачею охорони праці є зведення до мінімуму імовірність поразки чи захворювання працюючого з одночасним забезпеченням комфорту при максимальній продуктивності праці.

Комплекс організаційних і технічних заходів і засобів, що запобігають нещасні випадки, що виникають при впливі на працюючого небезпечних виробничих факторів, називається технікою безпеки. Виробнича санітарія містить у собі комплекс організаційних, гігієнічних і санітарно-технічних заходів, що запобігають виробничі шкідливості.

Турбота про охорону праці являється справою державної важливості. 14 жовтня 1992 року був прийнятий Закон України "Про охорону праці", у якому закріплені найважливіші положення в області охорони праці, і який поширюється на всіх суб'єктів підприємницької діяльності незалежно від форм власності. Основними пунктами в цьому законі є: стаття про гарантію прав громадян на охорону праці, стаття про відшкодування моральної шкоди, статті про охорону праці жінок, неповнолітніх, інвалідів.

У Законі "Про охорону праці" зазначені органи, покликані здійснювати нагляд і контроль за дотриманням законодавства про працю і правил по охороні праці. До них відносяться спеціально уповноважені на те державні органи й інспекції, що не залежать у своїй діяльності від адміністрації підприємств, установ, організацій і їхніх вищих органів.

Державне керування органами охорони праці здійснюють: Кабінет Міністрів України, державний комітет України по нагляду за охороною праці, міністерство й інші централізовані органи, місцеві ради народних депутатів і

їхні виконавчі комітети. Державний нагляд за дотриманням законодавства й інших нормативних актів по охороні праці здійснюють: державний комітет України по нагляду за охороною праці, державний комітет України по атомній і радіаційній безпеці, органи державного пожежного нагляду, органи і служби санітарно епідемічного нагляду. А трудові колективи в особі своїх уповноважених і профспілки здійснюють суспільний контроль за охороною праці.

Вищий нагляд за точним виконанням законів про працю міністерствами і відомствами, підприємствами, установами, громадянами і посадовими особами покладений на Генерального Прокурора України.

Велике значення в створенні здорових і безпечних умов праці має стандартизація. Вона дозволяє вжити діючих заходів по підвищенню технічного рівня й упорядкуванню розробки нормативно-технічної документації по безпеці праці. З 1972 року створюється Система стандартів по безпеці праці (ССБТ), що являє собою комплекс великого числа взаємозалежних стандартів, спрямованих на забезпечення безпеки праці. Ця система встановлює загальні вимоги і норми по видах небезпечних і шкідливих виробничих факторів, загальні вимоги безпеки до виробничого устаткування і процесів, вимоги до засобів захисту працюючих, методи оцінки безпеки праці.

4.2. Аналіз небезпечних і шкідливих виробничих факторів, характерних для приміщення фінансового відділу ВП ПУАЕС

Реальні виробничі умови характеризуються, як правило, наявністю деяких небезпечних і шкідливих виробничих факторів.

До небезпечних виробничих факторів відносяться фактори, вплив яких приводить до травми, фактори, що приводять до професійного захворювання працюючого. По природі дії небезпечні і шкідливі фактори за ДСТ 120.003-74 діляться на:

- фізичні;
- хімічні;
- біологічні;

- психологічні.

Впровадження обчислювальної техніки в усі сфери народного господарства привело до значного розширення парку ПЕВМ. У зв'язку з цим виникла проблема організації робочих місць користувачів ПЕВМ відповідно до вимог нормативів охорони праці.

Робота з персональним комп'ютером зв'язана з низкою шкідливих і небезпечних факторів, що можуть впливати на організм людини.

За даними Всесвітньої організації охорони здоров'я, професійна діяльність зв'язана з роботою на ПЕОМ може в окремих випадках приводити до порушень з боку зорового аналізатора, кістково-м'язовим (змушена поза) порушенням, нервово-емоційною напругою при роботі, шкірним захворюванням і ін. Високий відсоток (92%) складають скарги на зорове стомлення. Дослідження показали, що професія користувача ЕОМ являє собою модель розумової праці, що виконується в одноманітній позі в умовах обмеження загальної м'язової активності при рухливості рук, при високій напрузі зорових функцій і нервово-емоційній напрузі в умовах впливу різноманітних фізичних факторів. Дослідження умов праці показали, що на більшості підприємств користувачі ЕОМ працюють при несприятливому мікрокліматі, зниженій вологості повітря, нерациональному освітленні. Підвищений шум на робочому місці, унаслідок роботи комп'ютерів і принтерів, шкідливо впливає на органи слуху і на весь організм людини через центральну нервову систему, викликаючи швидку стомлюваність, головний біль, безсоння.

У зв'язку з використанням клавіатури можуть виникнути захворювання рук, спини, пліч і шиї.

Отже, підводячи підсумок, при роботі із ЕОМ можна виділити наступні фактори:

- фізичні – низькочастотне електромагнітне випромінювання моніторів, яскраві відблиски на екрані, підвищена яскравість світла монітора, недостатня освітленість робочого місця, запиленість повітря поблизу екрана дисплея через його електризацію, підвищення рівня шуму, підвищена напруженість

електричних і магнітних полів, підвищення рівня статичної електрики, небезпека поразки електричним струмом;

- психофізіологічні – тривалі статичні навантаження м'язів рук, шиї, спини, зап'ясть, пліч, ніг, постійна напруга зору, нервово-емоційні перевантаження (монотонність виконуваних дій, розумова перенапруга).

Установлено наступні санітарні норми і правила:

- мікроклімат у робочій зоні приміщення (до 2 м над рівнем підлоги) для легкої роботи (витрати енергії до 150 ккал/година);
- температура 17-22 С°;
- відносна вологість не більш 75%;
- швидкість руху повітря не більш 0,3 м/с.;
- припустимі коефіцієнти пульсації штучного освітлення в залежності від системи освітлення і характеру виконуваної роботи не повинні перевищувати 10-20%;
- освітленість при комбінованому освітленні 150-400 лк, при загальному освітленні – 200 лк, коефіцієнт запасу – 1.5 (для люмінесцентних ламп);
- рівень шуму – 50 дБА в кімнаті програмістів і до 65 дБА в машинному залі;
- гранично припустима напруженість електричної складової електромагнітного поля 10 В/м, магнітної складовий – 0,3 А/м;
- гранично припустима напруженість електростатичного поля при тривалому впливі на людину на робочому місці (до 1 години) – 60 кв/м.

4.3. Розрахунок системи освітлення для приміщення фінансового відділу ВП ПУАЕС

Рациональне освітлення приміщень – один з найбільш важливих факторів, від яких залежить ефективність трудової діяльності людини. Гарне освітлення необхідне для виконання більшості задач оператора. Для того, щоб спланувати раціональну систему освітлення, враховується специфіка робочого завдання, для якого створюється система освітлення, швидкість і точність, з яким це робоче

завдання повинне виконуватися, тривалість його виконання і різні зміни в умовах виконання робочих операцій.

Розрахунок освітленості будемо робити для приміщення, що має наступні характеристики:

- довжина приміщення $A = 16$ м;
- ширина приміщення $B = 6$ м;
- висота $h = 4$ м;
- число вікон 3;
- кількість робочих місць 3.

У приміщенні, де знаходиться робоче місце оператора використовується змішане освітлення, тобто сполучення природного і штучного освітлення. У якості природного – бічне освітлення через вікна. Штучне освітлення використовується при недостатнім природному освітленні. У даному приміщенні використовується загальне штучне освітлення.

Розрахунок його здійснюється по методу світлового потоку з урахуванням потоку, відбитого від стін і стелі. Нормами для даних робіт установлена необхідна освітленість робочого місця $E_n = 300$ лк (середня точність роботи з розрізнення деталей розміром від 1 до 10 мм).

Загальний світловий потік визначається по формулі

$$F_{\text{общ}} = \frac{E_n \cdot S \cdot z_1 \cdot z_2}{V} = \frac{300 \cdot 96 \cdot 1,5 \cdot 1,1}{0,34} = 139764,7 \text{ лм},$$

де E_n – нормована освітленість ($E_n = 300$ лк);

S – площа приміщення $S = A \cdot B = 16 \cdot 6 = 96$ м²;

$z_1 = 1,5$ – коефіцієнт, що враховує старіння ламп і забруднення світильників;

$z_2 = 1,1$ – коефіцієнт, що враховує нерівномірність висвітлення приміщення;

V – коефіцієнт використання світлового потоку; визначається в залежності від коефіцієнтів відображення від стін, стелі, робочих поверхонь, типів світильників і геометрії приміщення.

Виберемо з таблиці коефіцієнт використання світлового потоку по наступним даним:

- коефіцієнт відображення побіленої стелі $RP = 70\%$;

- коефіцієнт відображення від стін, пофарбованих у світлу фарбу $RCT=50\%$;

- коефіцієнт відображення від підлоги, покритого лінолеумом темного кольору $RP=10\%$;

- індекс приміщення:
$$i = \frac{A \cdot B}{h(A + B)} = \frac{16 \cdot 6}{4(16 + 6)} = 1,1.$$

Таким чином, коефіцієнт $V = 0,34$.

Для організації загального штучного освітлення виберемо лампи типу ЛБ40. Люмінесцентні лампи мають ряд переваг перед лампами накаливання: їхній спектр ближче до природного; вони мають велику економічність (більше світловіддача) і термін служби (у 10-12 разів). Поряд з цим мають і недоліки: їхня робота супроводжується іноді шумом; гірше працюють при низьких температурах; їх не можна застосовувати у вибухонебезпечних приміщеннях; мають малу інерційність.

Для нашого приміщення люмінесцентні лампи підходять.

Світловий потік однієї лампи ЛБ40 складає не менш $F_{\text{л}}=2810$ лм. Число N ламп, необхідних для організації загального освітлення визначається по формулі:

$$N = \frac{F_{\text{общ}}}{F_{\text{л}}} = \frac{139764,7}{2810} = 50.$$

Як світильники вибираємо ПВЛ-1, 2х40 Ут.

Таким чином, щоб забезпечити світловий потік $F_{\text{общ}}=139764,7$ лм треба використовувати 25 світильників по 2 лампи ЛБ40 у кожному.

Електрична потужність однієї лампи ЛБ40 $W_{\text{л}}=40$ Вт.

Потужність всієї освітлювальної системи

$$W_{\text{общ}} = W_{\text{л}} \cdot N = 40 \cdot 50 = 2000 \text{ Вт.}$$

4.4. Розробка заходів щодо забезпечення сприятливих умов праці при роботі з персональним комп'ютером у фінансовому відділі ВП ПУАЕС

Сьогодні фахівці в області ергономіки вже зрозуміли, що не можна знайти ідеальне положення, у якому можна перебувати і працювати протягом усього

робочого дня. Для більшості людей комфортабельним робочим місцем повинне бути таке, котре можна пристосувати не менш чим для двох позицій, при цьому положення крісла, монітора повинні щораз відповідати виконуваній роботі і звичкам. Багато хто вважають, що для роботи на комп'ютері більше підходять вертикальне і злегка похиле положення. Можливо, щоб крісло було злегка нахилене вперед.

Дисплей. Положення тіла звичайно відповідає напрямку погляду; дисплеї, розташовані занадто низько чи під неправильним кутом, є основними причинами сутулості. Відстань від дисплея до очей може варіюватися в залежності від характеру виконуваної роботи – 40-70 см. При роботі з текстом відстань від екрана до очей повинна лише небагато перевищувати відстань між книгою й очима і складати 40-45 див. Необхідно давати відпочинок очам і час від часу їх просто закривати. Не можна допускати, щоб очі увесь час були сфокусовані на одній відстані, працюючи з текстом, рекомендується установлювати великий шрифт. Дуже важливо, щоб екран монітора не мерехтів. Частота регенерації зображення повинна бути не менш 72 Гц, тому що при більш низькій частоті помітне мерехтіння, хоча люди з особливо чутливим зором можуть помітити його і при більш високій частоті. Їм доведеться підібрати графічну плату і монітор з частотою регенерації 85 Гц і вище.

Рекомендується встановлювати на екран монітора спеціальні скляні поляризаційні фільтри з заземленням. Необхідно уникати того, щоб термінал був звернений екраном убік вікна, оскільки інтенсивна освітленість області зору може затопити потоками світла очі і розмити зображення на сітківці. Якщо приходить сидіти поруч з вікном, то треба розташуватися під прямим кутом до нього, причому екран дисплея повинний бути перпендикулярний шибці – цим виключаються відблиски на екрані.

Для зниження впливу низькочастотного електромагнітного випромінювання монітора на ЕЛТ рекомендується вибирати монітор, що задовольняє стандарту MPR II, чи TCO 92-95. У протилежному випадку необхідно установити на екран захисний фільтр, найбільш сучасні моделі, затримують до 99 % електромагнітного випромінювання.

Форма спинки крісла повинна повторювати форму спини працюючого. Крісло необхідно установити на такій висоті, щоб не почувався тиск на куприк (крісло розташоване занадто низько) на стегна (крісло розташоване занадто високо). Фахівці з ергономіки, вважали, що кут між стегнами і хребтом повинний складати 90°, однак недавно проведені дослідження показали, що більшість людей переважно сидять трохи відкинувшись.

Клавіатуру необхідно установити так, щоб не треба було до неї тягтися і нахилитися вперед. При зміні положення тіла (наприклад, з вертикального на похиле) обов'язково треба перемінити положення клавіатури і дисплея. Може виявитися корисною регульована підставка клавіатури, завдяки якій можна без усякої напруги працювати з маніпулятором "миша". Можна поставити клавіатуру і на коліна. Руки при роботі повинні бути прямими в зап'ястях і зігнуті в ліктях приблизно під прямим кутом. Пальці також повинні бути злегка зігнуті. Удари по клавішах не повинні бути занадто сильними.

Зручна висота столу особливо важлива в тому випадку, коли на ньому розташовується клавіатура. Якщо в клавіатури немає підставки, а висоту столу не можна змінити (і він занадто високий), то треба вище підняти сидіння крісла, а під ноги підставити лавочку: чи що-небудь інше. Якщо стіл занадто низький, необхідно підкласти що-небудь під його ніжки.

Якщо при роботі часто приходиться дивитися на документи, треба установити підставку з оригіналом документа в одній площині і на одній, висоті з екраном. Якщо треба частіше дивитися на оригінал, чим на екран, необхідно повернути крісло й екран таким чином, щоб оригінал розміщався прямо перед очима, а комп'ютер ледве збоку.

Щогодини необхідно робити перерву в роботі. У цей час рекомендується робити вправи для зап'ясть.

5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.

5.1 Вступ.

За останні десятиліття кількість шкідливих, відходів підприємств настільки зросло, що природа поступово втрачає здатність до самоочищення. Стало очевидним, що будь-яка нова техніка і технологія, навіть високопродуктивна й економічно вигідна, але не задовольняючим сучасним вимогам охорони праці і навколишнього середовища, не є прогресивною і не відповідає основному напрямку технічного прогресу. Тому охорона навколишнього середовища в даний час здобуває першорядне значення у всіх областях діяльності, людини.

Сучасна політика держав в області охорони навколишнього середовища від забруднення ґрунтується на принципі "забруднив – плати". Уперше на міжнародному рівні даний принцип був висунутий Організацією економічного співробітництва і розвитку в 1972 році. З цього часу цей принцип став активно використовуватися в законодавчій практиці європейських і інших країн світу.

Для постійного контролю за станом навколишнього середовища застосовують різні методи, одним з них є екологічний моніторинг навколишнього середовища.

5.2 Моніторинг як система спостереження і контролю навколишнього середовища.

Моніторингом навколишнього середовища називаються регулярні, виконувані по заданій програмі спостереження природних середовищ, природних ресурсів, рослинного і тварини світу, що дозволяють виділити їхній стан і процеси що відбуваються в них під впливом антропогенної діяльності.

Під екологічним моніторингом варто розуміти організований моніторинг навколишнього середовища, при якому, по-перше, забезпечується постійна оцінка екологічних умов середовища життя людини і біологічних об'єктів (рослин, тварин, мікроорганізмів і т.д.), а також оцінка стану і функціональної цінності екосистеми, по-друге, створюються умови для визначення корегувальних дій у тих випадках, коли цільові показники екологічних умов не

досягаються.

У систему моніторингу повинні входити наступні основні процедури:

- виділення (визначення) об'єкта спостереження;
- обстеження виділеного об'єкта спостереження;
- складання інформаційної моделі для об'єкта спостереження;
- планування спостережень;
- оцінка стану об'єкта спостереження й ідентифікація його інформаційної моделі;
- прогнозування зміни стану об'єкта спостереження;
- представлення інформації в зручній для використання формі і доведення її до споживача.

- Основні задачі екологічного моніторингу:
- спостереження за джерелом антропогенного впливу;
- спостереження за фактором антропогенного впливу;
- спостереження за станом природного середовища під впливом факторів антропогенного впливу й оцінка прогнозованого стану природного середовища.

При розробці проекту екологічного моніторингу необхідна наступна інформація:

- джерело надходження забруднюючих речовин у навколишнє середовище – викиди забруднюючих речовин в атмосферу промисловими, енергетичними, транспортними й іншими об'єктами; скидання стічних вод у водяні об'єкти; поверхневі зливи забруднюючих і біогенних у поверхневі води суші і моря; внесення на земну поверхню і (чи) у ґрунтовий шар забруднюючих і біогенних речовин разом з добривами і ядохімікатами при сільськогосподарській діяльності; місця поховання і складування промислових і комунальних відходів; техногенні аварії, що приводять до викиду в атмосферу небезпечних речовин чи розливу рідких забруднюючих і небезпечних речовин, і т.д.;
- переноси забруднюючих речовин – процеси атмосферного переносу; процеси переносу і міграції у водяному середовищі;
- процеси ландшафтно-геохімічного перерозподілу забруднюючих

речовин – міграція забруднюючих речовин по ґрунтовому профілю до рівня ґрунтових вод; міграція забруднюючих речовин по ландшафтно-геохімічному сполученню з урахуванням геохімічних бар'єрів і біохімічних круговороті. Спостереження за цими процесами доцільно проводити періодично на спеціально виділеній системі пунктів: контрольні водозбори – катени – площадки – створи;

- дані про стан антропогенних джерел емісії – потужність джерела емісії і місце розташування його, гідродинамічні умови надходження емісії в навколишнє середовище.

5.3 Методи контролю забруднюючих речовин в об'єктах навколишнього середовища.

Сучасні методи контролю хімічних речовин, що забруднюють навколишнє середовище – це по суті фізико-хімічні методи. Іноді їх поєднують терміном "інструментальні методи аналізу". Дана тема величезна, тому ми розглянемо лише найбільш важливі з фізико-хімічних методів, що оптимально сполучають у собі цілий ряд якостей: високу точність і відтворюваність результатів аналізу, високу чутливість і, незважаючи на ці тверді вимоги, приступність апаратури і можливість швидкого її освоєння.

Газова хроматографія. В основу методу газової хроматографії покладений наступний принцип: аналіз суміші речовин у результаті розподілу компонентів між фазами, що не змиваються, одна з яких рухлива – інертний газ (азот, гелій і ін.), інша – нерухома (висококип'яча рідина чи тверда фаза).

Цей метод має два варіанти: газоабсорбційна і газорідинна хроматографія.

Поділ компонентів суміші відбувається в хроматографічному стовпчику. Хроматографічні стовпчики: набивні (довжина – 1-3 м, діаметр – 4 мм, матеріал – скло, сталь і ін.) і капілярні (довжина – до 50 м, матеріал – скло, кварц).

Вибір нерухомої фази (Нф). Ефективність стовпчика (здатність розділяти складні суміші на окремі компоненти) залежить від розміру часток, на які нанесена рідка фаза. Вона зростає при використанні однорідних часток малого розміру. Для стандартних набивних стовпчиків оптимальний розмір часток 0,12-

0,17 мм. Необхідно враховувати їхню близькість до аналізованих з'єднань. Для аналізу полярних компонентів застосовують полярні фази, для аналізу неполярних компонентів – менш полярні чи цілком неполярні.

Неполярні фази для газоадсорбційної хроматографії силікагель, оксид алюмінію, цеоліти, полімерні сорбенти (наприклад, полисорб, поропак і ін.).

Рухливі фази: азот, гелій, аргон, пари води.

Детектори. Історія розвитку газової хроматографії – це історія появи і розвитку детекторів для хроматографії. Застосовуються кілька типів детекторів.

1. Детектор теплопровідності (ДТП) чи катетометр. Принцип його дії заснований на розходженні теплопровідностей аналізованої речовини і газу-носія.

2. У детекторі іонізаційно плазмовому (ПІД чи ДПІ) використовується залежність електропровідності простору між електродами від числа іонізованих часток, що знаходяться в ньому, що утворюються у водневому полум'ї під дією термічних і окисних процесів при влученні в нього молекул аналізованої речовини.

3. Електронно-загарбний детектор (ЕЗД), чи детектор по захопленню електронів, як і ДІП, заснований на залежності електропровідності проміжку між електродами і числимо іонів, що знаходяться в цьому проміжку, що зв'язано з числом молекул, що надходять у детектор. Однак механізм і спосіб утворення іонів принципово відрізняються від такого у випадку ДІП – іони утворюються в результаті взаємодії молекул аналізованої речовини і потоку електронів у камері детектора в результаті бета-розпаду радіоактивної речовини.

4. Детектор термоіонний (ДТІ) за принципом дії аналогічний ДІП. Однак додатково у водневе полум'я безупинно надходить потік іонів лужних металів (калій, натрій, цезій). У їхній присутності різко зростає ефективність іонізації з'єднань, що містять азот, фосфор, хлор і ін. ДТІ застосовують для визначення ФОС і азотовмісних з'єднань.

5. Полум'яно-фотометричний детектор (ПФД) селективний і має підвищену чутливість стосовно з'єднань, що містять сірку.

Високоєфективна рідинна хроматографія (ВЕРХ) – хроматографічний

метод, що дозволяє розділити висококип'ячі рідини і (чи) тверді речовини, що важко або недоцільно визначати методом газорідинної хроматографії, наприклад полі циклічні ароматичні вуглеводні, амінокислоти, ПАВ, пестициди, лікарські препарати, вуглеводи й ін.

Тонкошарова хроматографія (ТШХ). Поділ відбувається на спеціальних пластинках для тонкошарової хроматографії. Нерухома фаза в ТШХ: силікагель, оксид алюмінію, іонообмінні смоли з добавками крохмалю і гіпсу.

Аналізовану суміш наносять на стартову лінію мікрошприцом чи мікропіпеткою. Пластинку чи папір з нанесеною пробою поміщають у закриту камеру, що містить розчинник, який переміщається по шару сорбенту (чи по паперові) під дією капілярних сил. Компоненти суміші переміщаються разом з розчинником з різними швидкостями. По закінченні поділу пластинку чи папір виймають з камери, випаровують розчинник, обробляючи струменем теплого повітря. Обумовлені речовини з'являються на хроматограмі у виді плям у результаті обробки спеціальним реактивом (наприклад, нігдрином при аналізі амінокислот) чи методом флюоресценції. Зміст аналізованого компонента пропорційно площі плям. Кількісну оцінку чи проводять безпосередньо на пластинці за допомогою планіметра, чи шляхом зняття пофарбованої плями з хроматограми екстракції речовини розчинником і визначення його зміст фотометричним чи методом за допомогою денситометра.

При аналізі катіонів колонку для поділу заповнюють сульфованими катіонами низкою ємності, а гнітючу колонком-аніонітом високої ємності. У якості елюентів використовують розчини HCl HNO_3 , гідро хлориду піридину й ін.

В останні роки розвивається іонна хроматографія без придушення фонового сигналу елемента і з різними способами детектування: фотометричний, атомно абсорбційний, іонометричний.

Достоїнства методу: низька межа визначення – 1–10 мг/мол, селективність, можливість одночасного визначення неорганічних і органічних іонів експресність, широкий діапазон обумовлених концентрацій.

Застосовують вітчизняний хроматограф "Колір-3006", кондуктометричний

детектор, мікропроцесор. Межа виявлення по хлориду натрію – 3,10 мг/мол.

Хроматомасс-спектрометрія (ХМС) – це в сутності газова хроматографія з мас-спектрометром як детектор (наприклад, МІ-1201). Даний метод дозволяє розшифровувати склад складних сумішей, що містять сотні невідомих компонентів, і визначати їх по одній пробі.

Полярографія. Це один з електрохімічних методів аналізу. Полярограма – залежність сили струму від величини прикладеної напруги на електроди. При цьому методі не відбувається фізичного поділу суміші на окремі компоненти. Як катод найчастіше застосовують ртутний електрод, що капає, (РКЕ), поверхня якого безупинна обновляється, що дозволяє одержувати полярограми і проводити аналіз з високою відтворюваністю результатів.

Пряме визначення можливе лише при наявності речовин, здатних відновлюватися на РКЕ: іони металів, органічні сполуки, що містять галоїд-, нітро-, карбонільні з'єднання, пероксиди, дисульфідів, і т.д. Це трохи обмежує можливості методу, однак при визначенні поліграфічних активних з'єднань дозволяє досягти високої селективності визначення без попереднього поділу складних сумішей на окремі компоненти.

Основні типи полярографії – постійна-токова (класична) і перемінно-токова. Остання має різні назви (підрозділи): у залежності від форми амплітуди перемінного струму – квадратно-хвильова.; у залежності від полярності електрода, що використовують як індикаторний, – катодна (відновлення) чи анодна (окислювання). Останню іноді називають вольтамперометрією. В анодній полярографії на відміну від катодної використовують тільки твердий електрод (наприклад, графітовий). Застосовують фоновий чи індиферентний електроліт (називаний просто – тло), тобто розчин кислоти, солі, буферний розчин більш складного складу, у якому розчиняють аналізовану пробу.

ВИСНОВКИ

В дипломній роботі було досліджено інформаційні потоки та розроблено систему документообігу ВП «Південноукраїнська АЕС». Були вивчені як спеціальні так і фундаментальні загально-технічні дисципліни.

Основною метою дипломної роботи є створення системи документообігу інформаційної системи ВП «Південноукраїнська АЕС».

Аналіз організаційної структури підприємства та основних видів діяльності виявили інформаційні потоки, на основі яких розроблені варіанти концепції інформаційної системи. В результаті функціонально-вартісної оцінки та з урахуванням вимог підприємства прийнято рішення щодо реалізації варіанту концепції побудови інформаційної системи для обліку та моніторингу замовлень клієнтів, як найбільш важливий напрямок діяльності.

За використанням технології UML розроблено статична та динамічна моделі інформаційної системи, які представлені у виді діаграм використання та переходів станів відповідно. Ці моделі є основою для розробки організаційного забезпечення і програмного забезпечення. Для прийняття рішень з інформаційного забезпечення використано аналіз організаційної структури підприємства, існуючої технології обробки замовлень. В результаті побудовано діаграму та розроблені логічна та фізична моделі.

В дипломній роботі інформаційна система розглядається як частина корпоративної інформаційної системи всього підприємства. Ці обставини мають вплив на рішення з технічного забезпечення, які передбачають склад технічних засобів для функціонування інформаційної системи міського підрозділа та засоби для підключення до корпоративної мережі підприємства.

Реалізація проекту інформаційної системи дає можливість підняти рівень обслуговування клієнтів та розширити базу клієнтів підприємства і, таким чином, позначиться не тільки на підвищенні економічних показників фірми, продуктивності праці, а й на швидкодії виконуваних процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Гарсиа-Молина Г., Ульман Д. Д., Уидом Д., Системы баз данных. Полный курс., М.: «Вильямс», 2004. – 1083 с.
- 2 Михайлюк В.О., Изотов В.І., Штейн П.В. Методичні вказівки до виконання розділу з питань безпеки життєдіяльності та цивільної оборони у дипломних роботах (проектах) студентів. – Миколаїв: НУК, 2008. – 56 с.
- 3 ГОСТ 12.0.003-74. Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация. – Введ 01.01.1976.
- 4 Виробниче освітлення та його розрахунок: Навчальний посібник. – Миколаїв: УДМТУ, 2001. – 84 с.
- 5 «Приемы объектно-ориентированого проектирования. Паттерны проектирования» Э.Гамма, Р.Хелн, Р.Джонсон, Дж. Влиссидес – спб. Питер, 2001 – 368с.
- 6 Цивільна оборона: Навчальний посібник для студентів вищих навчальних закладів / В.М. Шоботов.- 2-ге вид., перероб. – К.: Центр навчальної літератури, 2006. – 438 с.
- 7 Трофимов С.А. CASE-технологии. Практическая работа в Rational Rose. – Бином-Пресс, 2002. – 288с.
- 8 Проектування інформаційних систем. Навчальний посібник (затв. МОН України) // Литвин В.В., Шаховська Н.Б. / За ред. Пасічника В.В. – Львів. Магнолія 2006, 2011. – 380с.
- 9 Рэнд Моримото, Майкл Ноэл и др. Microsoft Windows Server 2008. Полное руководство: Пер. с англ. – М.: Издательский дом "Вильямс", 2011. – 1456 с.
- 10 ЄСКД «Загальні вимоги до текстових документів» (ДОСТ 2.105-95).

Технічне завдання

1 Загальні відомості

Підсистема інформаційної системи розробляється для ВП «Південноукраїнська АЕС», яка призначена для оптимізації ведення документообігу.

2 Підстави для розробки

Проект ІС розробляється в рамках дипломного проектування зі спеціальності ІУСТ.

3 Цілі і призначення розробки:

- поліпшення умов праці;
- підвищення умов обслуговування клієнтів;
- підвищення ефективності вирішення задач.

4 Вимоги до ІС:

4.1 Вимоги до функціональних характеристик:

Система документообігу повинна виконувати наступні функції:

- прийом та реєстрація замовлень;
- моніторинг і контроль замовлень;
- організація єдиного центру збору інформації;
- централізоване ведення архіву;
- автоматична обробка й аналіз фінансової інформації та прийняття рішень про подальші дії;
- збереження даних, занесення первинної інформації в БД (довідкова інформація, права доступу користувачів);
- підготовка робочих станцій: технічна модернізація робочих станцій, установка програмного забезпечення;

- підготовка комп'ютерної мережі: підключення всіх станцій у комп'ютерну мережу підприємства.

4.2 Вимоги до надійності

Повинна бути забезпечена надійність збереження інформації при наступних аварійних ситуаціях:

- відключення електроживлення;
- вихід з ладу серверної станції;
- вихід з ладу локальної комп'ютерної мережі.

4.3 Вимоги до програмної та інформаційної сумісності

Необхідно операційна система MS Windows XP/Vista/7 або Linux, встановлена система управління базами даних Microsoft Office Access.

4.4 Вимоги до складу та параметрів технічних засобів:

- сервер на платформі (DDRIII 2048Mb, SATA 250Gb);
- серверне програмне забезпечення (Windows XP);
- клієнтські платформи (Intel Celeron DC T3500 2.1GHz, FSB 667MHz, SLC 1Mb);
- мережне середовище (1 Гб/сек Ethernet).
- 700 Mb вільного місця на жорсткому диску, 2 Gb ОЗУ.

5 Вимоги до програмної документації

У програмну документацію повинні входити наступні документи:

- інструкція з експлуатації для користувача;
- інструкція з формування і ведення бази даних.

6 Календарний план створення системи

Календарний план з розробки підсистеми зображений у таблиці А.1.

Таблиця А. 1 - Календарний план розробки системи

Назва етапів	Термін виконання етапів проекту
Вивчення і опис предметної області	06.09.-06.10.2018
Формування вимог до ІС	07.10.-20.10.2018
Розділ з розробки проектних рішень по системі	21.10.-10.11.2018
Розробка робочої документації	11.11.-27.11.2018

7 Порядок контролю та приймання

Приймання створеної системи здійснюється представниками виконавця і замовника відповідно до технічного завдання. Складається двосторонній акт, що стає підставою для розрахунків.

Керівництво користувача

1 Установка ПЗ інформаційної системи

Для користувача системи встановлення додаткового ПЗ не потрібне. Для вимог до апаратних засобів суттєвим є лише можливість доступу до Internet. Також повинен бути встановлений сучасний Web-браузер.

2 Ініціювання роботи з системою

Ініціювання роботи з ІС відбувається шляхом введенням з Web-браузера адреси сервера, що містить інтерфейс для роботи з системою. Робота починається з авторизації користувача в системі. Для цього відразу після запуску програми у вікні авторизації потрібно ввести ім'я, пароль та натиснути кнопку "Увійти". Для скасування авторизації і негайного виходу з системи потрібно натиснути кнопку "Скасувати". Після успішної процедури ідентифікації з'являється головне вікно програми з меню керування її модулями.

Користувачами системи є адміністратор, менеджери, співробітники у бригадах.

3 Інструкція для використання функцій інтерфейсу

Після авторизації з'являється головне вікно з можливістю вибору наступних команд: реєстрація нового замовлення, формування замовлення, моніторинг замовлень та вихід з системи.

Реєстрація нового замовлення

При виборі пункту **Реєстрація нового замовлення** відкривається нове вікно, в якому потрібно ввести певні поля нового замовлення, потрібні для його реєстрації в системі – опис, вид. Крім того, потрібно вказати клієнта, який є ініціатором його подання. Для цього можна вибрати ПІБ клієнта зі списку на сторінці або натиснути кнопку Додати клієнта.

При цьому з'являється нове вікно, де потрібно ввести реквізити клієнта – ПІБ, адресу та телефон та натиснути кнопку Зберегти. Система зберігає нового клієнта в базу даних та повертає активність попередньому вікну.

Тепер нового клієнта можна вибрати зі списку. При натисненні кнопки Зареєструвати, замовлення зберігається до бази даних і активність передається до головного вікна.

Формування замовлення

Коли замовлення зареєстроване, над ним можна провести операцію формування. Для цього потрібно в головному вікні програми вибрати пункт **Формування замовлення**, вибрати зі списку зареєстроване замовлення і натиснути кнопку Формувати.

В новому вікні пропонується ввести наступні дані про замовлення: ціна замовлення, місце виконання, дата виконання. Також необхідно вибрати бригаду та матеріали для виконання замовлення.

Бригада

Для вибору бригади потрібно вибрати її в списку наявних бригад. Для змінення або видалення вибраної бригади слід натиснути відповідні кнопки Змінити або Видалити. Також можна створити нову за допомогою кнопки Створити бригаду. При цьому відкривається нове вікно, в якому необхідно заповнити опис даної бригади та кількість співробітників.

Учасники бригади

Далі потрібно додати в бригаду співробітників. Це можна зробити за допомогою кнопки Вибрати співробітника, після натиснення якої з'явиться вікно зі списком співробітників. При виборі ПІБ співробітника і натисненні кнопки ОК з'являється попереднє вікно бригади з новим записом в таблиці співробітників нижче на сторінці. Також потрібно ввести посаду учасника бригади. Для змінення або видалення вибраного учасника бригади слід натиснути відповідні кнопки Змінити або Видалити поряд з таблицею учасників.

Співробітник

Якщо співробітника немає в списку, слід вибрати пункт Додати нового співробітника. У новому вікні потрібно ввести його відомості: ПІБ, ідентифікаційний номер, адресу, телефон, посаду. Після натиснення кнопки Зберегти, система повернеться до вибору співробітників, і в списку буде наявний новий запис. Для змінення або видалення вибраного співробітника слід натиснути відповідні кнопки Змінити або Видалити.

Поставки матеріалів

Поставки матеріалів для даного замовлення та їх постачальники відображаються в таблиці поставок матеріалів на сторінці формування замовлення. Для змінення або видалення вибраної поставки, слід натиснути відповідні кнопки Змінити або Видалити. Щоб додати нову поставку для замовлення, потрібно натиснути кнопку Додати поставку. З'явиться вікно, в якому потрібно вибрати зі списків матеріал та певного постачальника. Також потрібно ввести ціну поставки, дату поставки, номер рахунку та код накладної.

Матеріали

Якщо певного матеріалу немає, можна його створити шляхом натиснення на кнопку Додати матеріал поряд зі списком матеріалів. В новому вікні необхідно ввести його атрибути: найменування, опис, одиниці вимірювання, марку. Для змінення або видалення вибраного матеріалу, слід натиснути відповідні кнопки Змінити або Видалити.

Постачальник

Якщо дане найменування матеріалу присутнє, але потрібен інший постачальник, то вибираємо його назву зі списку каталогу матеріалів, і при цьому інші поля стають заповненими та неактивними. Після цього необхідно вибрати зі списку постачальника. Для змінення або видалення вибраного постачальника слід натиснути відповідні кнопки Змінити або Видалити.

Якщо постачальника не зазначено у списку, слід натиснути на кнопку Додати постачальника. У новому вікні потрібно ввести наступні дані: назву постачальника, факс, телефон, адресу, умови оплати, примітку. Після натиснення кнопки ОК, новий постачальник стає наявним у списку на сторінці матеріалу.

Після введення всієї необхідної інформації про замовлення, потрібно натиснути на кнопку Виконати замовлення, після чого буде внесено до бази даних усі введені поля та записи з інших таблиць, що пов'язані з даним замовленням.

Моніторинг замовлень

Для отримання інформації про наявність та хід виконання замовлень, в головному вікні слід вибрати пункт **Моніторинг замовлень**. Далі відкриється таблиця, в якій можна побачити стан виконання нових замовлень. Кожне замовлення відображається в одному рядку даної таблиці. Його основні атрибути відображаються в стовбцях. Можуть з'являтися або бути відсутніми наступні атрибути: дата виконання та місце виконання. Відповідно до процесу виконання можуть змінюватися й інші поля.

Для редагування або видалення замовлення потрібно виділити відповідний рядок та натиснути кнопку Змінити або Видалити. Порядок появи вікон та змін у них аналогічний пункту Формування замовлення.

Вихід з системи

Для виходу з системи користувачеві достатньо вибрати пункт **Вихід з системи**. При цьому з'явиться вікно авторизації. Також вихід можливий безпосередньо на етапах формування замовлення, вибору бригад, замовлення матеріалів.

Програма і методика випробувань

Об'єкт випробувань

Об'єктом випробувань є система документообігу інформаційної системи ВП «Південноукраїнська АЕС».

Цілі випробувань

Цілі випробувань: перевірка придатності системи до використання по призначенню, її відповідність заданим вимогам і документації; перевірка працездатності і правильності роботи системи.

Вимоги до системи

Список вимог до системи приведений у Додатку А пояснювальної записки.

Порядок проведення випробувань:

При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливо використання нових чи додаткових тестів).

Першим етапом проведення випробувань програмного продукту буде проведення візуального контролю. Візуальний контроль - це перевірка програм “за столом”, без використання комп'ютера. На першому етапі візуального контролю здійснюється читання програми, при чому особлива увага надається коментарям і їх відповідності тексту програми, умовам в операторах умовного вибору (If, Switch) і циклу, складним логічним виразам, можливості не завершення ітераційних циклів (While, Do).

Другим етапом проведення випробувань програмного продукту буде проведення статичного контролю. Статичний контроль - це перевірка програми по її тексту (без виконання) за допомогою інструментальних засобів. Оскільки написання програмного продукту відбувалося за допомогою інтерпретатора

РНР, то як засоби проведення статичного контролю, будуть вибрані саме вище перелічені компілятори.

Третім етапом проведення випробувань програмного продукту буде проведення функціонального тестування. При функціональному тестуванні програма розглядається як “чорний ящик ” (тобто її текст не використовується). Відбувається перевірка відповідності поведінки програми її зовнішньої специфікації. Критерієм повноти тестування в цьому випадку є перебір всіх можливих значень вхідних даних, що в даному випадку нездійсненно. Оскільки вичерпне функціональне тестування неможливе, може йтися про розробку методів, що дозволяють підбирати тести не “всліпу ”, а з великою вірогідністю виявлення помилок в програмі.

Методика випробувань

В ході проведення першого етапу тестування програмного продукту – візуального контролю, виявлено, що всі коментарі в програмі відповідають тексту програми, умови в операторах умовного вибору і циклах поставлені коректно, складні логічні вирази побудовані коректно і відповідають задачам, що пред'являються до програмного комплексу, можливостей не завершення ітераційних циклів не існує.

Під час проведення другого етапу тестування програмного продукту – статичного контролю, з використанням інструментальних засобів компілятора РНР, було виявлено, що код усіх програмних модулів, який компілюється, не має жодного попередження першого, другого і третього рівнів, а також не має жодної помилки. При написанні модулів програми, були виконані усі умови об'єктно-орієнтованого програмування на мові РНР. При виконанні статичного контролю виявлено:

- 1) у текстах програм всіх програмних модулів не використовуються неініціалізовані змінні;
- 2) у програмі немає описів елементів, змінних, процедур, міток і файлів, що не використовуються у тексті;
- 3) у програмі немає фрагментів коду, що ніколи не виконуються;

4) у тексті програми немає змінних, значення яких ніколи не використовується після їх ініціалізації початковими значеннями;

Таблиця Б.1 – Тестування створення проекту

Дія	Результат
Вводимо порожнє значення в поле «Project name»	Відмова в створенні проекту. Вивід повідомлення про помилку. Пропозиція коректно ввести ім'я проекту.
Вводимо коректне значення в поле імені.	Створення проекту.

Таблиця Б.2 – Тестування створення альбому

Дія	Результат
Вводимо порожнього значення в поле «Name»	Відмова у створенні звіту. Вивід повідомлення про помилку. Пропозиція коректно ввести ім'я.
Залишаємо невиданим поле «Style»	Відмова у створенні звіту. Вивід повідомлення про помилку. Пропозиція вибрати один з стилів для альбому.
Вибираємо поле «Style» залишаємо невиданим поле «Page»	Відмова у створенні звіту. Вивід повідомлення про помилку. Пропозиція вибрати кількість сторінок у звіті.
Коректно вказуємо усі параметри.	Звіт створюється успішно.

Третім етапом тестування програмного продукту є функціональне тестування. Тестування функції введення початкових даних ми робимо, використовуючи стратегію чорного ящика. Для кожної з введених записів, у

базу даних перевіряється правильність їх подальшого використання модулями програми. Тестування триватиме до тих пір, поки всі дані використовуватимуться за призначенням.

Тестування програмних модулів проводиться з метою виявлення найбільшого числа помилок в програмних модулях. Для тестування програмних модулів застосовуються методи чорного і білого ящика.

Тестування проводиться методом "чорна скриня" із застосуванням методів еквівалентного розбиття, припущень про помилку і аналізу граничних значень.

«Чорний ящик» є складною моделлю кібернетичної системи, в якій дотримується різноманітність. Він лише тоді є задовільною моделлю системи, коли містить таку кількість інформації, яка відображає різноманітність системи. Можна передбачити, що ніж більше число обурень діє на входи моделі системи, тим більшу різноманітність повинен мати регулювальника.

Маніпулюючи тільки з входами і виходами, можна проводити певні дослідження. На практиці завжди виникає питання, наскільки гомоморфізм «чорного» ящика відображає адекватність його моделі, що вивчається, тобто як повно в моделі відбиваються основні властивості оригінала.

Опис будь-якої системи управління в часі характеризується картиною послідовності її станів в процесі руху до мети, що стоїть перед нею. Перетворення в системі управління може бути або взаємно-однозначним і тоді воно називається ізоморфним, або лише однозначним, в один бік. У такому разі перетворення називають гомоморфним.

Тестування по цьому методу йде під управлінням вхідних/вихідних даних. У таблицях приведені класи еквівалентності для тестування методом "чорна скриня" програмних модулів.

**Таблиця Б3 – Класи еквівалентності для модуля
«Введення логіну та паролю»**

№	Вхідні умови	Правильний клас	Не правильний клас
1	Логін співробітника	Рядок не пустий	Рядок пустий
2	Пароль співробітника	Рядок не пустий	Рядок пустий

Таблиця Б4 – Класи еквівалентності для модуля «Менеджер»

№	Вхідні умови	Правильний клас	Не правильний клас
1	Логін співробітника	1)Рядок не пустий 2)Є ключ шифрування	1)Рядок пустий 2)Відсутній ключ шифрування
2	Пароль співробітника	1)Рядок не пустий 2)Є ключ шифрування	1)Рядок пустий 2)Відсутній ключ шифрування

Таблиця Б5 – Класи еквівалентності для модуля «Адміністратор»

№	Вхідні умови	Правильний клас	Не правильний клас
1	Логін співробітника	1)Рядок не пустий 2)Є ключ шифрування	1)Рядок пустий 2)Відсутній ключ шифрування
2	Пароль співробітника	1)Рядок не пустий 2)Є ключ шифрування	1)Рядок пустий 2)Відсутній ключ шифрування

Таблиця Б6– Класи еквівалентності для модуля

«Приймання замовлення»

№	Вхідні умови	Правильний клас	Не правильний клас
1	Реєстрація замовлення	1)Рядок не пустий 2)Є вхідні дані	1)Рядок пустий 2)Відсутні вхідні дані
2	Формування замовлення	1)Рядок не пустий 2)Є умови формування замовлення	1)Рядок пустий 2)Відсутні умови формування замовлення

Програма випробувань

Тести складаються замовником і/або його представниками. При цьому перевіряється функціонування системи з нормальними і помилковими даними і подіями.

Для визначення переліку вхідних даних для кожного тесту можна скористатися формами для введення відповідних даних. У формах зазначені необхідні поля.

Вихідні дані чи результат при правильних даних повинні відповідати виконанню функції, для тестування якої створений тест; при неправильних даних – відмовлення в доступі до системи чи повідомлення про помилку.

Фрагмент тексту програми

```

<?php

class user
{

    function dispatch($action)
    {
        $smarty = Core::lib('sm');
        $user = Core::user();
        if ($user)
            $smarty->assign('user', $user);

        $profile = DAOFactory::getInstance()->getAccountDAO()-
>getAccountByScreenname($action);

        if (!$profile)
        {
            Core::lib('http')->redirect('/search/members/?s=' .
urlencode($action));
        }

        switch (@$_GET['sub'])
        {
            case 'groups':
                $this->groups($profile);
                break;
            case 'contacts':
                $this->contacts($profile);
                break;
            case 'images' :
                $this->images($profile);
                break;
            case 'lightbox' :
                $this->lightbox($profile['id']);
                break;
            case 'portfolio':
                $this->portfolio($profile['id']);
                break;
            case 'journal' :
                $this->journal($profile);
                break;
            case 'gallery' :
                $this->gallery($profile);
                break;
            default:
                $this->profile($profile);
        }
    }

    function portfolio($id)
    {
        Core::get('rss')->userportfolios($id);
    }

    function lightbox($id)
    {
        if(@$_GET['sub2'] == 'rss')
            return Core::get('rss')->userfavorites($id);
        $smarty = Core::lib('sm');
    }
}

```

```

global $DOMAIN;

$grid = @$_COOKIE[self::GRID_STYLE_COOKIE];
if ($grid != 'f' && $grid != 'm')
    $grid = 'f';

$sort = @$_COOKIE[self::SORT_LIGHTBOX_COOKIE];
if ($sort != 'rating' && $sort != 'date' && $sort != 'title')
    $sort = 'date';

$cookieTime = time() + self::YEAR_TIME;
setcookie(self::GRID_STYLE_COOKIE, $grid, $cookieTime, '/', $DOMAIN);
setcookie(self::GRID_STYLE_COOKIE, $grid, $cookieTime, '/');
setcookie(self::SORT_LIGHTBOX_COOKIE, $sort, $cookieTime, '/', $DOMAIN);
setcookie(self::SORT_LIGHTBOX_COOKIE, $sort, $cookieTime, '/');

50);
$items = DAOFactory::getInstance()->getImageDAO()->queryFavorites($id, 0,

$images = &$items->getData();
self::addLinks($images);

$smarty->assign( array(
    'grid' => $grid,
    'sort' => $sort,
    'items_json' => $items->toJSON(false),
    'owner_id' => $id
));
$smarty->display('dqr:user/lightbox.tpl');
}

function journal($profile)
{
    $smarty = Core::lib('sm');

    $user = Core::user();
    $http = Core::lib('http');

    $dao = &DAOFactory::getInstance()->getPortfolioDAO();
    $ret = $dao->getAccountImages($profile['id'], 0, 100, true, $user ?
$user['id'] : null, false, true, 'id', false);
    self::transform($ret);

    $smarty->assign('user', $user);
    $smarty->assign('owner', $profile);
    $smarty->assign('images_json', $ret->toJSON(false));

    $smarty->assign('months_json', json_encode(Core::lib('datehelper')-
>getMonths()));
    $smarty->assign('days_json', json_encode(Core::lib('datehelper')-
>getDays()));

    $smarty->display('dqr:user/journal.tpl');
}

function gallery($profile)
{
    $smarty = Core::lib('sm');
    $smarty->assign('gallery_id', 'null');
    $smarty->assign('account_id', $profile['id']);

    $user = Core::user();

    $gallery = DAOFactory::getInstance()->getGalleryDAO()-
>getAccountGallery($profile['id'], $user ? $user['id'] : null);
    if (!$gallery)

```

```

        $template = 2;
    else
        $template = $gallery['template'];

    $smarty->assign('template_id', $template);
    $smarty->assign('rand', rand());
    $smarty->assign('gallery', $gallery);
    $smarty->assign('harmony', Core::lib('gallery_colour_harmonies')-
>getHarmony($gallery['harmony']));
    $smarty->display('dqr:gallery/index.tpl');
}

function profile(&$profile, $template = 'dqr:user/index.tpl')
{
    $user = Core::user();
    $factory = &DAOFactory::getInstance();
    $accountDao = $factory->getAccountDAO();
    $profile = $accountDao->getProfile($profile['screenname'], $user ?
$user['id'] : null);
    if (!$profile)
    {
        Core::lib('http')->redirect('/search/members/?s=' .
urlencode($profile['screenname']));
    }

    $profile['viewerId'] = $user ? $user['id'] : null;

    $accountDao->hit($profile['id'], Core::lib('tracker')->sessionId);

    $smarty = Core::lib('sm');

    // feed
    $feed = array(
        array('date' => time(),
            'message' => 'Something weird happened'),
        array('date' => time(),
            'message' => 'Something weird happened again'),
    );

    self::addLinks($profile['highlights']);

    if (Core::user())
    {
        require_once(ROOT . 'dao/contactgroupfilter.php');
        $filter = new ContactGroupFilter();
        $filter->setAccountId($profile['viewerId']);
        $smarty->assign('contactgroups', $factory->getContactGroupDAO()-
>query($filter));
    }

    // tags
    $tags = Core::lib('taghelper')->getAccountTags($profile['id'], $user ?
$user['id'] : null);
    // for SEO keywords, list all tags
    $tagwords = array();
    foreach ($tags->getData() as $t)
        $tagwords[] = htmlspecialchars($t['name']);
    $tagwords = implode(' ', $tagwords);
    $messengers = Core::lib('profile_messenger')->get();
    foreach($profile['messengers'] as &$m)
        $m['type'] = $messengers[ $m['type'] ]['name'];

    $smarty->assign(array(
        'person' => array(

```

```

        'id' => $profile['id'],
        'avatar' => Core::lib('avatarhelper')->user($profile, false),
        'viewerid' => $profile['viewerId'],
        'type' => Core::lib('photographertype')-
>lookup($profile['type'])->getName(),
        'firstname' => $profile['firstname'],
        'lastname' => $profile['lastname'],
        'displayname' => $profile['displayname'],
        'userpic' => $profile['userpic'],
        'screenname' => $profile['screenname'],
        'description' => $profile['description'] ,
        'status' => $profile['status'],
        'zipcode' => $profile['zipcode'],
        'address' => $profile['address'],
        'city' => $profile['city'],
        'country' => $profile['country'],
        'countryname' => $profile['countryname'],
        'updated' => $profile['updated'],
        'gender' => $profile['gender'],
        'birthdate' => $profile['birthdate'],
        'trueyear' => date('Y', $profile['birthdate']) > 1905,
        'cameras' => $profile['cameras'],
        'lens' => $profile['lens'],
        'interests' => $profile['interests'],
        'messaging' => $profile['messaging'],
        'website' => ($profile['website'] == 'http://') ? false :
$profile['website'],
        'company' => $profile['company'],
        'studio' => $profile['studio'],
        'studio_description' => $profile['studio_description'],
        'photographers' => $profile['photographers'],
        'networks' => $profile['networks'],
        'other_services' =>
self::transform_services($profile['photoservices']),
        'networks' => self::transform_networks($profile['networks'])
    ),
    'contacts' => self::transform_accounts($profile['contacts']),
    'incontacts' => $profile['incontacts'],
    'groups' => self::transform_groups($profile['groups']),
    'short_months_json' => json_encode(Core::lib('datehelper')-
>getShortMonths()),
    'lightbox' => self::transform($profile['lightbox']),
    'highlights' => $profile['highlights'],
    'latestpics' => self::transform($profile['latestpics']),
    'gallery_enabled' => $profile['gallery_enabled'],
    'feed' => new ResultSet($feed, count($feed), 0),
    'tagwords' => $tagwords
));

$imageutils = Core::lib('imageutils');

foreach ($profile['comments'] as &$c)
{
    $c['small'] = $imageutils->url($c, 't');
    $c['avg_rank'] = ceil($c['avg_rank']);
}
$smarty->assign('comments', $profile['comments']);

$portfolios = $profile['latestportfolios'];
$data = &$portfolios->getData();
foreach ($data as &$portfolio)
{
    if ($portfolio['cover'])
        $portfolio['cover'] = array('small' => $imageutils-
>url($portfolio['cover'], 't'),

```

```

'square' => $imageutils-
>url($portfolio['cover'], 'q'));
    }

    // global $MEMCACHED_HOST, $MEMCACHED_PORT;
    // if ($mc = @memcache_connect($MEMCACHED_HOST, $MEMCACHED_PORT))
    // {
    //     $suid = @memcache_get($mc, $profile['id']);
    // }
    // else
    //     $suid = false;

    $smarty->assign('isonline', $suid);
    $smarty->assign('latestportfolios', $portfolios);

    $smarty->display($template);
}

function contacts(&$profile)
{
    $user = Core::user();
    $smarty = Core::lib('sm');

    $cookie = @$_COOKIE[self::CONTACTS_VIEW_COOKIE];
    if ($cookie)
        $viewmode = $cookie;
    else
        $viewmode = 'medium';

    $p_viewmode = @$_GET['view'];
    if ($p_viewmode)
        $viewmode = $p_viewmode;

    if ($viewmode != 'short' && $viewmode != 'table' && $viewmode !=
'medium')
        $viewmode = 'medium';

    self::set_contacts_view_cookies($viewmode);

    $page = intval(@$_GET['page']);
    if ($page)
        $page -= 1;

    switch ($viewmode)
    {
        case 'short':
            $pagesize = self::CONTACTS_PER_PAGE_SHORT;
            break;
        case 'table':
            $pagesize = self::CONTACTS_PER_PAGE_TABLE;
            break;
        default:
            $pagesize = self::CONTACTS_PER_PAGE_MEDIUM;
    }

    $dao = &DAOFactory::getInstance()->getContactGroupDAO();

    $group = @$_GET['group'];

    require_once(ROOT . 'dao/contactgroupfilter.php');
    $filter = new ContactGroupFilter();
    $filter->setName($group);
    $filter->setAccountId($profile['id']);

```

```

$filter->setStart($page * $pagesize);
$filter->setCount($pagesize);
$filter->setNumImages(5);

$smarty->assign('contacts', self::transform_accounts($dao-
>queryContacts($filter, ContactGroupDAO::ORDER_DATE)));

$smarty->assign('viewmode', $viewmode);
$smarty->assign('pagesize', $pagesize);
$smarty->assign('group', $group);
$smarty->assign('profile', $profile);

$filter->setName(null);
$filter->setStart(0);
$filter->setCount(10);
$filter->setNumImages(0);

if (!@$_GET['widget'])
{
    // last updated, most used, most active
    $contacts = $dao->queryContacts($filter,
ContactGroupDAO::ORDER_POPULARITY);

    $smarty->assign('popularcontacts',
self::transform_accounts($contacts));
    $smarty->assign('activecontacts', self::transform_accounts($dao-
>queryContacts($filter, ContactGroupDAO::ORDER_ACTIVITY)));

    // list of groups
    $filter = new ContactGroupFilter();
    $filter->setAccountId($profile['id']);
    $smarty->assign('contactgroups', $dao->query($filter));

    $sids = array();
    foreach($contacts->getData() as $d)
    {
        $sids[$d['id']] = 1;
    }
    Core::lib('taghelper')->getAccountTags(array_keys($sids), $user ?
$user['id'] : null, 50);

    $smarty->display('dqr:user/contacts.tpl');
}
else
    $smarty->display('dqr:user/contactsview.tpl');
}

function groups(&$profile)
{
    $smarty = Core::lib('sm');

    $viewmode = @$_COOKIE[self::GROUPS_VIEW_COOKIE];

    $p_viewmode = @$_GET['view'];
    if ($p_viewmode)
        $viewmode = $p_viewmode;

    if ($viewmode != 'short' && $viewmode != 'table' && $viewmode !=
'medium')
        $viewmode = 'medium';

    self::set_groups_view_cookies($viewmode);

    $page = intval(@$_GET['page']);
    if ($page)
        $page -= 1;
}

```

```

switch ($viewmode)
{
    case 'short':
        $pagesize = self::GROUPS_PER_PAGE_SHORT;
        break;
    case 'table':
        $pagesize = self::GROUPS_PER_PAGE_TABLE;
        break;
    default:
        $pagesize = self::GROUPS_PER_PAGE_MEDIUM;
}

$dbao = &DAOFactory::getInstance()->getAccountGroupDAO();

$user = Core::user();

$smarty->assign('groups',
    self::transform_groups($dbao-
>getAccountGroups($profile['id'],
    $page *
    $pagesize,
    AccountGroupDAO::ORDER_DATE,
    $viewmode
    == 'medium' ? 5 : 0,
    $user ?
    $user['id'] : null)));
    $smarty->assign('viewmode', $viewmode);
    $smarty->assign('pagesize', $pagesize);

    if (!@$_GET['widget'])
    {
        // last updated, most used, most active
        $smarty->assign('populargroups', self::transform_groups($dbao-
>getAccountGroups($profile['id'], 0, 10, AccountGroupDAO::ORDER_POPULARITY)));
        $smarty->assign('activegroups', self::transform_groups($dbao-
>getAccountGroups($profile['id'], 0, 10, AccountGroupDAO::ORDER_ACTIVITY)));
        $smarty->assign('ratedgroups', self::transform_groups($dbao-
>getAccountGroups($profile['id'], 0, 10, AccountGroupDAO::ORDER_RATE)));
        $smarty->display('dqr:user/groups.tpl');
    }
    else
        $smarty->display('dqr:user/groupsview.tpl');
}

function images($profile)
{
    if( @$_GET['sub2'] == 'rss')
        Core::get('rss')->userimages($profile['id']);
    else
        Core::get('portfolio')->execute(null, $profile['screenname'],
        @$_GET['sub2']);
}

private static function set_contacts_view_cookies($cookie)
{
    global $DOMAIN;
    setcookie(self::CONTACTS_VIEW_COOKIE, $cookie, time() + 365 * 60 * 60 *
24, '/', $DOMAIN);
    // TODO: temporary line to work with localhost
    setcookie(self::CONTACTS_VIEW_COOKIE, $cookie, time() + 365 * 60 * 60 *
24, '/');
}

```



```

private static function set_groups_view_cookies($cookie)
{
    global $DOMAIN;
    setcookie(self::GROUPS_VIEW_COOKIE, $cookie, time() + 365 * 60 * 60 * 24,
'/', $DOMAIN);
    // TODO: temporary line to work with localhost
    setcookie(self::GROUPS_VIEW_COOKIE, $cookie, time() + 365 * 60 * 60 * 24,
'/' );
}

private static function transform_services($rs)
{
    Core::lib('site');
    $services = array();
    foreach ($rs as $service)
    {
        $helper = sitehelper::lookup($service['service_id']);
        if (!$helper)
            continue;
        $url = $helper->normalize($service['name']);
        if ($url)
            $services[] = $url;
    }
    return $services;
}

private static function transform_networks($rs)
{
    Core::lib('site');
    $networks = array();
    foreach ($rs as $network)
    {
        $helper = sitehelper::lookup($network['network_id']);
        if (!$helper)
            continue;
        $url = $helper->normalize($network['name']);
        if ($url)
            $networks[] = $url;
    }
    return $networks;
}

private static function transform_groups($rs)
{
    $data = &$rs->getData();
    foreach ($data as &$group)
    {
        $group['userpic'] = Core::lib('avatarhelper')->group($group, false);
        // display localized number of accounts in group
        $group['num_accounts'] = sprintf(ngettext('%d member in group', '%d
members in group', $group['num_accounts']), $group['num_accounts']);
        if (@$group['images'])
        {
            foreach ($group['images'] as &$i)
            {
                $i['url'] = 'http://' . SQUARE_THUMBNAIL_BUCKET .
'.s3.amazonaws.com/' . $i['name'];
                if (!$i['title'])
                    $i['title'] = $i['original_name'];
            }
        }
    }
    return $rs;
}

```

```

    }

    private static function transform_accounts($rs)
    {
        $data = &$rs->getData();
        foreach ($data as &$account)
        {
            $account['userpic'] = Core::lib('avatarhelper')->user($account,
false);
            if (@$account['images'])
            {
                foreach ($account['images'] as &$i)
                {
                    self::transform_image($i);
                }
            }
        }
        return $rs;
    }

    private static function transform_image(&$image)
    {
        if (!$image)
        {
            return;
        }
        if (!$image['title'])
        {
            $image['title'] = $image['original_name'];
        }
        $image['url'] = 'http://' . SQUARE_THUMBNAIL_BUCKET .
'.s3.amazonaws.com/' . $image['name'];
    }

    private static function transform(&$rs)
    {
        $data = &$rs->getData();
        Core::lib('texthelper');
        $imageutils = Core::lib('imageutils');

        foreach ($data as &$image)
        {
            if (!$image['title'])
            {
                $image['title'] = $image['original_name'];
            }
            $image['state'] = 'r';
            $image['big'] = $imageutils->url($image, 's');
            $image['small'] = $imageutils->url($image, 't');
            $image['square'] = $imageutils->url($image, 'q');
            unset($image['state']);
            unset($image['i_l']);
            unset($image['i_b']);
            unset($image['i_m']);
            unset($image['i_s']);
            unset($image['i_t']);
            unset($image['i_q']);
            unset($image['i_h']);
            $image['description'] = Core::lib('texthelper')->
article($image['description']);
        }
        return $rs;
    }

    private static function addLinks(&$images)
    {
        $imageutils = Core::lib('imageutils');
        $options = array('links' => true);
        foreach( $images as &$i )
        {
            if(isset($i['name']))
            {
                $imageutils->transform_image_object($i, $options );
            }
        }
    }

```

```

}

{capture assign=head}
<script src='/js/jquery.simplemodal.js' type='text/javascript'></script>
<script type="text/javascript" src="/js/jquery.postjson.js"></script>
<script type="text/javascript" src="/js/swfobject.js"></script>
<link href="/css/modal.css" rel="stylesheet" type="text/css" media="screen" />
{/capture}
{capture assign=body}
{if $incontacts eq 3 and !empty($contactgroups)}
<div id="addcontactdialog" style="display:none; height:100%;">
    <div class="popupBlock">
        <h1>{t}Add friend{/t}</h1>

        <div class="grad_block">
            <h2>{t}What group should it be?{/t}</h2>
            {assign var=cgdata value=$contactgroups->getData() }
            <ul class="popup_settings" id="iset_list_cg">
                {section name=k loop=$cgdata}
                    <li><div class="btn-checkbox-{if
$smarty.section.k.first}on{else}off{/if}">
                        <input type="radio" id="cg_id" name="cg_id"
value="{ $cgdata[k].id}" {if $smarty.section.k.first} checked="checked" {/if} />
                        </div>{ $cgdata[k].name|escape:'html' }
                    </li>
                {/section}
            </ul>
        </div>
        <div class="popupButtons">
            <input type="button" id="closedialog" value="{t}CANCEL{/t}"
class="btn_grey" onclick="$modal.close()">
            <input type="button" id="submit_contactgroup" value="{t}OK{/t}"
class="btn_orange">
        </div>

    </div>
</div>
{/if}
{include file="dqr:common/tooltip.tpl"}
{/capture}
{if $person.viewerid eq $person.id}
{include file="dqr:user/header.tpl" activemenu='mydqr'
title="`$person.displayname`, `$person.type` - DarQroom.com, high quality online
photo sharing" metakeywords="darkroom darqroom `$person.displayname` photo
sharing hosting album portfolio images `$tagwords`"
metadescription="`$person.displayname`, `$person.type` shares his photography
passion and talent on DarQroom"}
{else}
{include file="dqr:community/header.tpl" title="`$person.displayname`,
`$person.type` - DarQroom.com, high quality online photo sharing"
metakeywords="darkroom darqroom `$person.displayname` photo sharing hosting album
portfolio images `$tagwords`" metadescription="`$person.displayname`,
`$person.type` shares his photography passion and talent on DarQroom"}
{/if}

{include file="dqr:user/profile.tpl"}

{include file="dqr:common/footer.tpl"}

<script type="text/javascript" src="/js/user/user.profile.js"></script>
<div id="getStart">
    <div id="getStar_left">
        <div id="blockProfile">
            {assign var=name value="`$person.firstname`
`$person.lastname`"}

```

```

        {if $isonline}<b
class="online_stat">{t}Online{/t}</b>{else}<b
class="offline_stat">{t}Offline{/t}</b>{/if}
        <h1>{$name|strip|escape:'html'}</h1>
        <div id="profile_avatar">
            <p>
                {if $person.userpic }
                
                {else}
                
                {/if}
            </p>
            {if isset($user) && $user.id != $person.id}
            <p>
                <ul class="rating star{$person.avg_rank}"
style="margin-left:auto; margin-right:auto;">
                    <li id="rate-1"><a
href="javascript:;"></a></li>
                    <li id="rate-2"><a
href="javascript:;"></a></li>
                    <li id="rate-3"><a
href="javascript:;"></a></li>
                    <li id="rate-4"><a
href="javascript:;"></a></li>
                    <li id="rate-5"><a
href="javascript:;"></a></li>
                </ul>
            </p>
            {/if}
        </div>

        <div id="profile_info">
            {if $person.status }
            <p
class="profile_rank">{$person.type|escape:'html'}</p>
            {/if}
            {if $person.zipcode}
            <p
class="profile_location">{$person.zipcode|escape:'html'}</p>
            {/if}
            {if $person.city and $person.countryname }
            <p
class="profile_location">{$person.city|escape:'html'} /
{$person.countryname|escape:'html'}</p>
            {elseif $person.city}
            <p
class="profile_location">{$person.city|escape:'html'}</p>
            {elseif $person.countryname}
            <p
class="profile_location">{$person.countryname|escape:'html'}</p>
            {/if}
            {if $person.updated }
            <p class="profile_lastupdate">{t}Last update{/t}
            <span class="utc" format="%Y-%m-%d" >{$person.updated}</span></p>
            {/if}
            <br />
            {if $person.gender }
            <p class="white">{if $person.gender eq
1}{t}Male{/t}{else}{t}Female{/t}{/if}</p>
            {/if}
            {if $person.birthdate }
            <p> {t}Birthdate{/t} <span class="utc white"
format="{if $person.trueyear}%Y-%m-%d{else}%d %M{/if}"
>{$person.birthdate}</span> </p>
            {/if}

```

```

        <br />
        {if $person.description }
        <p
class="profile_desc">{$person.description|strip|escape:'html'}</p>
        {/if}
        <br />
        {if $person.website }
        <p>{t}Website:{/t} <a href="{ $person.website}"
target="_blank">{$person.website}</a></p>
        {/if}
        {if !empty($person.networks) or
!empty($person.other_services) }
        <p class="profile_links">See: <span
class="white">
                {section name=net loop=$person.networks}
                <a href="{ $person.networks[net] }"
target="_blank">{$person.networks[net]|escape:'html'}</a>
                {/section}
                {section name=site
loop=$person.other_services}
                <a href="{ $person.other_services[site] }"
target="_blank">{$person.other_services[site]|escape:'html'}</a>
                {/section}
        </span></p>
        {/if}
        <br />
        {if !empty($person.cameras) }
        <p>{t}My cameras:{/t} <span class="white">
                {section name=cam loop=$person.cameras}
                { $person.cameras[cam]|escape:'html' }
                {/section}</span></p>
        {/if}
        {if !empty($person.lens) }
        <p>{t}My lens:{/t} <span class="white">
                {section name=len loop=$person.lens}
                { $person.lens[len]|escape:'html' }
                {/section}
        </span></p>
        {/if}
        {if $person.interests }
        <p>{t}Interests:{/t} <span
class="white">{$person.interests|escape:'html'}</span></p>
        {/if}
        {if $person.company }
        <p>{t}Company:{/t} <span
class="white">{$person.company|escape:'html'}</span></p>
        {/if}
        {if $person.studio }
        <p>{t}Studio:{/t} <span
class="white">{$person.studio|escape:'html'}</span></p>
        {/if}
        {if $person.studio_description}
        <p>{t}Studio description:{/t} <span
class="white">{$person.studio_description|escape:'html'}</span></p>
        {/if}
        {if $person.photographers}
        <p>{t}Favorite photographers:{/t}<span
class="white">
                {section name=p loop=$person.photographers}
                { $person.photographers[p] }
                {/section}
        </span></p>
        {/if}

        {if $person.messengers}

```

```

        <p>{t}Instant messengers:{/t}<span class="white">
            {section name=mm loop=$person.messengers}
                <p>{$person.messengers[mm].type}
{$person.messengers[mm].name}</p>
            {/section}
        </span></p>
    {/if}
</div>

    <div class="clear">&nbsp;</div>
</div>

{*
    <div class="hr">&nbsp;</div>

    <div id="blockSales">
        
    </div><!-- // END of ArtworkSales --> *}

    <div class="hr">&nbsp;</div>

    <div id="blockLatestPics">
        <div class="wallbutton_warp">{include
file='dqr:common/wallbutton.tpl' objectid='latestwall'
query="portfolio/images&account_id='{$person.id}'&recursive=true&sort_by=id&sort_as
c=false"}</div>
        <h2>
            
            <a href="/user/{$person.screenname}/images/rss"></a>
        </h2>

        <div class="greyblock">
            <b class="r1">&nbsp;</b>
            <b class="r2">&nbsp;</b>
            <div class="greyblockContent">
                <div id="latestpics">
                    {if $latestpics->getCount()}
                    <ul id="latestpics">
                        {assign var=data value=$latestpics-
>getData()}
                        {section name=i loop=$data}
                            {include
file="dqr:common/smallimagebox.tpl" image=$data[i]
link="/portfolio/detail/`$data[i].portfolio_id`#`$data[i].id`"}
                            {/section}
                        </ul>
                    {else}
                        <div class="empty_message">{t}No
photos{/t}</div>
                    {/if}
                </div>
                {if $latestpics->getCount()}<a
href="/search/images?s={$person.screenname|escape:'url'}&a=1&di=30&du=d&ha=1"
class="lnk_viewmore">{t}more{/t}</a>{/if}
                <div class="clear">&nbsp;</div>
            </div>
            <b class="r2">&nbsp;</b>
            <b class="r1">&nbsp;</b>
        </div>
    </div><!-- // END of My Latest Photos -->

    <div class="hr">&nbsp;</div>

```

```

        <div id="blockLighbox">
            <div class="wallbutton_warp">{include
file='dqr:common/wallbutton.tpl' objectid='favoritewall'
query="image/queryfavorites&account_id=`$person.id`&sort_by=id&sort_asc=false"}</
div>
                <h2>
                <a
href="/user/{$person.screenname}/lightbox/rss"></a>
            </h2>
            <div class="greyblock">
                <b class="r1">&nbsp;</b>
                <b class="r2">&nbsp;</b>
                <div class="greyblockContent">
                    <div id="lightbox">
                        {if $lightbox->getCount() }
                        <ul id="lightbox">
                            {assign var=data value=$lightbox-
>getData() }
                                {section name=i loop=$data}
                                    {include
file="dqr:common/smallimagebox.tpl" image=$data[i]
link="/portfolio/detail/`$data[i].portfolio_id`#`$data[i].id`" }
                                    {/section}
                                </ul>
                                {else}
                                <div class="empty_message">{t}No
images{/t}</div>
                                {/if}
                            </div>
                            {if $lightbox->getCount() }<a
href="/user/{$person.screenname}/lightbox"
class="lnk_viewmore">{t}more{/t}</a>{/if}
                            <div class="clear">&nbsp;</div>
                        </div>
                        <b class="r2">&nbsp;</b>
                        <b class="r1">&nbsp;</b>
                    </div>
                </div><!-- // END of My Favorites -->

                <div class="hr">&nbsp;</div>

                <div id="blockHonoredPics">
                    <h2> <a href="/user/{$person.screenname}/portfolio"></a></h2>

                    <div id="portfolios_list" class="holder">
                        <ul>
                            {assign var=data value=$latestportfolios-
>getData() }
                            {section name=p loop=$data}
                                <li>
                                    <div class="portfolio_border">
                                        <a href="/portfolio/{$data[p].id}">
                                            {if $data[p].cover}
                                                
                                                {else}
                                                
                                            {/if}
                                        </a>

```

```

        </div>
        <div class="portfolio_title"><a
href="/portfolio/{ $data[p].id }">{ $data[p].name|truncate:20:'...':true|escape:'html' }</a></div>

        <div class="portfolio_author"><a
href="/user/{ $data[p].screenname }">{ $data[p].displayname|truncate:20:'...':true|escape:'html' }</a></div>

        <div class="portfolio_launch"><a
target="_blank" href="/gallery/{ $data[p].id }">{t}Launch Gallery
view{/t}</a></div>

    </li>
    {/section}
</ul>
<div class="clear">&nbsp;</div>
</div>

<div class="hr">&nbsp;</div>

<div id="blockGroups">
    {if $person.viewerid eq $person.id}
    <a href="/group/create" class="orange_button"
style="float:right;">{t}Create group{/t}</a>
    {/if}
    <h2></h2>
    {include file="dqr:common/groups.tpl"
url="/user/{ $person.screenname }/groups"}
</div>

</div>

<div id="getStar_right">

    <div id="profile_menu">
    { if $incontacts eq 2 }
    <h2>{t}Friend{/t}</h2>
    <ul id="profile_menu_actions">
    {elseif $incontacts eq 3 }
    <ul id="profile_menu_actions">
    <li id='friend_status'><a href="javascript:void(0)"
onclick="show_contacts_dialog();">{t}Add Friend{/t}</a></li>
    {else}
    <ul id="profile_menu_actions">
    {/if}
    {if isset($user) }<li><a
href="/messages/compose/u{ $person.id }">{t}Send a message{/t}</a></li>
    <li><a href="#" onclick="return false">{t}Share
this{/t}</a></li>
    {/if}
    </ul>
    <div class="clear">&nbsp;</div>
    <ul id="profile_menu_buttons">
    {if $gallery_enabled}
    <li><a target="_blank"
href="/user/{ $person.screenname }/gallery"></a></li>
    {/if}
    <li class="left"><a
href="/user/{ $person.screenname }/images"></a></li>
    {if $person.viewerid eq $person.id}
    <li class="right"><a href="/my"></a></li>
    {/if}

```



```

{*
                                <li><a
href="/user/{ $person.screenname }/journal"></a></li> *}
                                </ul>
                                </div>

{*
                                <div id="blockNewsFeed">
                                    <a href="#" class="ico-rss"></a>

                                    <h2>{t}News feed{/t}</h2>
                                    <div class="greyblock">
                                        <b class="r1">&nbsp;</b>
                                        <b class="r2">&nbsp;</b>
                                        <div class="greyblockContent">

                                            {assign var=data value=$feed->getData()}
                                            {section name=f loop=$data}
                                                <dl {if $smarty.section.f.index eq 0}
class="first" {/if}>
                                                    <dt class="utc" format="%y-%m-%d"
>{$data[f].date}</dt>
                                                    <dd><a
href="#">{$data[f].message|truncate:30:'...':true|escape:'html'}</a></dd>
                                                    </dl>
                                                {/section}

                                                    <div class="clear">&nbsp;</div>
                                                    </div>
                                                    <b class="r2">&nbsp;</b>
                                                    <b class="r1">&nbsp;</b>
                                                </div>
                                            </div>
                                        *}

                                <div id="blockFastPictures">
                                    <div class="greyblock">
                                        <b class="r1">&nbsp;</b>
                                        <b class="r2">&nbsp;</b>
                                        <div class="greyblockContent">
                                            {if $highlights.rated and $highlights.popular and
$highlights.latest}

                                                <ul id="fast_pictures_list">
                                                    {assign var=pic value=$highlights.rated}
                                                    {if !empty($pic)}
                                                        <li>
                                                            <h3></h3>
                                                            <a
href="/portfolio/detail/{ $pic.portfolio_id }#{ $pic.id }" class="at"
title="{ $pic.title|strip|escape:'html' } /
{ $pic.description|strip|truncate:200:'...':true|escape:'html' }"></a>
                                                            <a
href="/user/{ $person.screenname }/images/loved">{t}view
more{/t}&nbsp;&nbsp;&nbsp;></a>
                                                            </li>
                                                        {/if}
                                                    {assign var=pic value=$highlights.popular}
                                                    {if !empty($pic)}
                                                        <li>
                                                            <h3></h3>
                                                            <a
href="/portfolio/detail/{ $pic.portfolio_id }#{ $pic.id }" class="at"

```

```

title="{ $pic.title|strip|escape:'html' } /
{ $pic.description|strip|truncate:200:'...':true|escape:'html' }"></a>
<a
href="/user/{ $person.screenname}/images/popular">{t}view
more{/t}&nbsp;&nbsp;&nbsp;&gt;</a>
</li>
{/if}
{assign var=pic value=$highlights.latest}
{if !empty($pic)}
<li>
<h3></h3>
<a
href="/portfolio/detail/{ $pic.portfolio_id}#{ $pic.id}" class="at"
title="{ $pic.title|strip|escape:'html' } /
{ $pic.description|strip|truncate:200:'...':true|escape:'html' }"></a>
<a
href="/user/{ $person.screenname}/images/latest">{t}view
more{/t}&nbsp;&nbsp;&nbsp;&gt;</a>
</li>
{/if}
</ul>
{else}
<div class="empty_message">{t}No pictures
uploaded yet{/t}</div>
{/if}

<div class="clear">&nbsp;</div>
</div>
<b class="r2">&nbsp;</b>
<b class="r1">&nbsp;</b>
</div>
</div>

<div id="blockArticles">
<h2></h2>
<div class="greyblock">
<b class="r1">&nbsp;</b>
<b class="r2">&nbsp;</b>
<div class="greyblockContent">
{if $comments}
<ul class="list">
{section name=f loop=$comments}
<li>
{if $comments[f].title}
{assign var=tt
value=$comments[f].title|truncate:32:'...':true|escape:'html'}
{else}
{assign var=tt
value=$comments[f].original_name|truncate:32:'...':true|escape:'html'}
{/if}
<a class="at"
title=" / &lt;div
class=&quot;tt_comment&quot; &gt;&lt;b&gt;{ $tt|escape:'html' }&lt;/b&gt;&lt;img
src=&quot;{ $comments[f].small}&quot; /&gt;&lt;br/&gt;&lt;div class=&quot;rating4
star{ $comments[f].avg_rank}&quot; &gt;&lt;/div&gt;&lt;/div&gt;"

href="/portfolio/detail/{ $comments[f].pid}#{ $comments[f].id}">
<span class="utc" format="%Y-
%m-%d">{ $comments[f].date}</span> -
{ $comments[f].text|truncate:28:'...':true|escape:'html' }

```

```

        </a>
    </li>
</section>
</ul>
<div class="empty_message">{t}Images are not
commented yet{/t}</div>
    </if>
    <div class="clear">&nbsp;</div>
</div>
<b class="r2">&nbsp;</b>
<b class="r1">&nbsp;</b>
</div>
</div>

{include file="dqr:common/tagscloud.tpl" resultset=$tags}

<div id="blockContacts">
    <h2></h2>
    <div class="greyblock">
        <b class="r1">&nbsp;</b>
        <b class="r2">&nbsp;</b>
        <div class="greyblockContent">
            {if $contacts->getCount()}
            <ul class="blockPopularList">
                {assign var=data value=$contacts->getData()}
                {section name=c loop=$data}
                {include
file="dqr:common/smallmemberbox.tpl" profile=$data[c]}
                </section>
            </ul>
            <a href="/user/{ $user.screenname}/contacts"
class="lnk_viewmore">{t}view more{/t}</a>
            {else}
            <div class="empty_message">{t}No
contacts{/t}</div>
            </if>
            <div class="clear">&nbsp;</div>
        </div>
        <b class="r2">&nbsp;</b>
        <b class="r1">&nbsp;</b>
    </div>
</div>

</div>
<div class="clear">&nbsp;</div>
</div>
<script type="text/javascript">
var SHORT_MONTHS = {$short_months_json};
var account_id = '{$person.id}';
</script>

function show_contacts_dialog()
{
    $('#addcontactdialog').modal({
        close: false
    });

    $('#submit_contactgroup').bind('click',
        function(){
            $.postJSON('/1.0/contactgroup/addcontacts', { account_id: account_id,
id: $(".btn-checkbox-on").find('input').val() },
            function(data){

```

```

        if(data[account_id] == 'ACCOUNT_ADDED'){
            $.modal.close();
            $('#friend_status').html('Friend');
        }
    }
    );
}
);
}

$(document).ready(function(){
    $('.blockMembers .blockTabs a').bind('click', function(e) {
        var mode = param(this.href, 'members') || 'latest';
        $('.blockMembers .blockTabs a').removeClass('act');
        $(this).addClass('act');
        $('#mholder').load('/community/members/' + mode, null, function(text,
status, req) {
            autotooltips();
        });
        e.stopPropagation();
        e.preventDefault();
        return false;
    });

    $('#blockGroups .blockTabs a').bind('click', function(e) {
        $('#blockGroups .blockTabs a').removeClass('act');
        $(this).addClass('act');
        var mode = param(this.href, 'groups') || 'latest';
        $('#gholder').load('/community/groups/' + mode, null, function(text,
status, req) {
            autotooltips();
        });
        e.stopPropagation();
        e.preventDefault();
        return false;
    });

    $('#iset_list_cg div').click(function() {
        $('#iset_list_cg div').each(function() {
            if ($(this).hasClass('btn-checkbox-on')) {
                $(this).removeClass('btn-checkbox-on');
                $(this).addClass('btn-checkbox-off');
            }
            $(this).find('input').removeAttr('checked');
        });
        $(this).removeClass('btn-checkbox-off');
        $(this).addClass('btn-checkbox-on');
        $(this).find('input').attr('checked', 'checked');
    });

    $('.rating a').click( function(e) {
        var rate = $(this).parent().attr('id').substring('rate-'.length);

        $.postJSON('/1.0/account/rate', {id: account_id, rate: rate}, function(o)
{
            if (o == 1)
                $('.rating').removeClass('star0 star1 star2 star3 star4
star5').addClass('star' + rate);
        });
    });
});

```

autotooltips();